Improved Damage Tolerance of Composites by Promoting Multiple Delaminations - Quasi-Static Loading

Master Thesis

Group DMS4 23C
Design of mechanical systems
Aalborg University

02-06-2022

AALBORG UNIVERSITY
STUDENT REPORT

The Faculty of Engineering and Science
Mechanical Engineering and Production
Fibigerstræde 16
DK - 9220 Aalborg East
http://www.engineering.aau.dk/

Title:

Improved Damage Tolerance of Composites by Promoting Multiple Delaminations - Quasi-static Loading

Project:

Master thesis

Project period:

01-02-2022 - 02-06-2022

Project group:

DMS4 - 23C

Participants:

Jonas Kyneb Wulff Andersen
Rasmus Lohmann Bjerg
Lasse Højmark Larsen

Supervisor:

Ass. Prof Esben Lindgaard

Pages: 87
Appendix: 24
Handed in 02-06-2022

Synopsis:

In this master thesis it is investigated how multiple delaminations in fiber composites can be promoted by using toughening/weakening patches in order to improve damage tolerance. This is done by first investigating state-of-the-art methods for toughening the interface between the layers in composite materials, to determine the ones that qualifies for being used as patches in DCB specimens.
Four different arc-length solvers are presented and implemented to solve the inherent nonlinear behavior associated with delamination problems. A parameter study has been done alongside the implementation of the arc-length solvers. The parameter study are conducted on the parameters of the patches and their placement in the DCB specimens. The findings from the parameter study are used to manufacture DCB specimens for the experimental tests, where four different patches are tested. Specifically, two interleaving methods and a stitching method are used as toughening patches. Additionally, DCB specimens with a weakening patch is also used. The results show that the weakening patch promotes multiple delaminations, which improves the fracture resistance by approximately 50 %, whereas it was not possible to promote multiple delaminations with toughening patches.

Resumé

Denne kandidatafhandling undersøger, hvordan multiple delamineringer i fiberkompositter kan fremprovokeres ved at forstærke eller svække interlaminære områder, med det formål at forbedre skadetolerancen.
Dette gøres ved at undersøge state-of-the-art og brainestorme forskellige metoder til at forstærke eller svække områder i fiberkompositten.
Den efterfølgende undersøgelse tager udgangspunkt i to dele, som er udført sideløbende. Dette omfatter udarbejdelse og implementering af numeriske bue-længde løsere til anvendelse i et Finite Element Program. Fire bue-længde løsere er undersøgt og implementeret. To af disse baserer sig på en geometriske begrænsnings ligninger, hvilket kan give problemer i tilfælde, hvor lokal materiale ustabilitet forekommer. Dette kan løses, ved i stedet at definere bue-længde løseren baseret på en ligning, der relaterer sig til mængden af energien, der dissiperes i modellen. Dette anvendes i de to resterende bue-længde løsere, der er implementeret. Alle fire bue-længde løsere er valideret op mod en flytningskontrolleret finite element løsning. Robustheden af den ene løser er dog begrænset.
Sideløbende med bue-længde løserne er der lavet et parameterstudie. Dette parameterstudie udføres på paramenterne for de interlaminære områder, der forstærkes eller svækkes i fiberkompositten, samt deres placering i såkaldte Double Cantilever Beam (DCB) emner. Resultaterne fra parameterstudiet anvendes til at fremstille DCB emner til de eksperimentelle forsøg. Der fremstilles en type, hvor et interlaminært område svækkes, og tre hvor et interlaminært område forstærkes. Specifikt anvendes der en tynd PTFE-film til at svække det interlaminære område, hvor der ved forstærkningen af det interlaminære område anvendes to forskellige mikrofiber måtter og en syningsmetode.
Resultaterne fra de eksperimentelle forsøg viser, at ved at svække de interlaminære områder kan der fremprovokeres multiple delamineringer, hvilket forbedrer sejheden med ca. 50 %. Det var derimod ikke muligt at fremprovokere multiple delamineringer med de to forskellige mikrofiber måtter eller syningsmetoden.

Preface

This master thesis is written during the 4th semester 2022 in the Design of Mechanical Systems master's program at Aalborg University with supervising by associate professor Esben Lindgaard.

The authors want to express their gratitude to the supervisor of this thesis, Associate Professor Ph.D. Esben Lindgaard, for his inspiration, guidance and help during this thesis. The authors also want to thank Post.Doc Simon Mosbjerg Jensen and Ph.D. Guillem Gall Trabal for assisting when needed.

Further, the author want to thank the staff from the Materials and Production workshop at Aalborg University for assisting in manufacturing DCB specimens.

Reading guide:
Citations in this report is done using the Harvard method [Surname, year]. A citation before a punctuation accounts for the sentence, with a citation after the punctuation accounting for the paragraph. The bibliography is written in the end of the report followed by the appendices.

Figures and pictures in this report are either drawn or captured by the authors, unless stated otherwise.

At the end of the report, several appendices have been inserted with alphabetical numbering. The appendices are referred in the report and indicated by letter designation.

Jonas K. W. Andersen	Rasmus L. Bjerg	Lasse H. Larsen

By signing this document, each member of the group confirms participation on equal terms in the process of writing the project. Thus, each member of the group is responsible for all contents in the project.

Contents

1 Delamination of composites

The application of fiber reinforced, polymer-matrix (FRP) composite materials were at first introduced in the aerospace and aeronautics industry, but has since spread to other industries (Turon, 2007). A FRP composite is in general described as individual lamina stacked together to form a laminate, with the desired properties. Often composite materials are associated with high performance structures and nowadays also counts industries within naval engineering, civil engineering, automobile, high end sporting equipment, etc.

Generally, the reason for utilizing composite materials in structural components are its often superior properties, compared to conventional materials like e.g. steel. Especially, a high specific strength and stiffness, and the possibility for directionally tailored properties are the main benefits. Additionally, good fatigue properties, corrosion resistance, wear resistance etc. are often associated with composite materials. (Jones, 1999)

However, using composite materials in a structural design also comes at cost as it greatly complicates the design phase. This is due to e.g. added coupling terms, new failure modes, directional dependent properties, etc. This further complicates the prediction of the service life of a component.

Nonetheless, the advantages seems to outbalance the added complexity, as composites in many cases has either maintained or increased their importance within the aforementioned industries, since their introduction.

In addition to the introduction of advanced material, like composites, the demand for more efficient, lighter or larger structures, have over time led to new design approaches. These are needed in order to comply with new requirements, material and manufacturing cost while securing the structural integrity of components. According to Braga et al. (2014) three design approaches are generally used to handle fatigue in the design of high performance structures. These are safe-life, fail-safe and damage tolerant design, respectively.

Although the safe-life and fail-safe design approach are still in use, the most commonly applied design approach today, when considering high performance structures, are based on the damage tolerant design approach. This is mainly due to the limitations in the safe-life and fail-safe design philosophies which in some cases have lead to disastrous premature failure due to unforeseen failure modes. In other cases these have lead to unnecessary high safety factors or a uneconomically short service life (Boller and Buderath, 2007).

Damage tolerance is a measure of a structures or materials capability to sustain a load and/or functionality, when damaged. Thus, the structure is designed to safely operate even though damage is present. Further, in this design philosophy damage are allowed to grow throughout the service period, but must not become critical before it is detected (Braga et al., 2014). Thus, the damage tolerant design provides a guidance for, when to replace or

repair damaged components, while also taking into account the risk of final failure under continued service (Goyal, 2002).

The main advantages of this approach is the possibility of utilizing the materials much more efficiently, which may lead to more optimized structures (Boller and Buderath, 2007). Furthermore, the damage tolerant design philosophy, allows for extending the service lifetime of structures, compared to the safe-life and fail-safe design approaches, by monitoring the amount of damage throughout its service life (Braga et al., 2014).

However, implementation of a damage tolerant design philosophy requires thorough understanding of the materials including among others strength, fatigue properties, potential damage mechanisms and modes. Also advanced analysis methods that can predict damage initiation and propagation as a function of the in-service loads are needed. These modelling techniques are used along with conditional non-destructive inspection (NDI), live monitoring of damage in the structure and repair to ensure that non of the existing damage becomes critical in between two subsequent inspections. (Tavares and de Castro, 2018)

Additionally, a structure can be made more damage tolerant by using damage tolerant materials. Such materials should to a high degree exhibit stable crack growth. This implies that the load level needed to promote unstable crack growth, must be substantially higher, than the load level needed to initiate a crack. Thereby, materials which shows a toughening behaviour as a crack develops and grows is especially suited for damage tolerant designs. (McGugan et al., 2015)

1.1 Failure Mechanism in Composite Materials

Based on the introduction of the damage tolerant design approach in the previous section, it is clear that a thorough understanding of potential failure modes is required in order to use the damage tolerant design approach.

Failure in composite materials can overall be divided into two categories, intra-lamina and inter-laminar failure, respectively. Intra-lamina failure refers to failure modes within a lamina, while inter-laminar failure refers to failure in the interface between two adjoined laminas. However, it is often not enough to treat these types of failure separately as they often interact with each other (Goyal, 2002).

Often encountered intra-lamina failure modes are fiber failure, matrix failure and matrix/fiber interface failure or debonding. Fiber failure can occur due to tensile fracture, crushing or local compressive buckling or kinking. Matrix failure is typically associated with matrix cracking. However, also effects such as material degradation due to radiation or moisture can cause matrix failure. (Turon, 2007).

Inter-laminar failure is normally caused by loss of adhesion between adjoined laminas and typically happens due to the weak interface strength, which is commonly found in composite materials. This type of failure is referred to as delamination and is one of the most frequently seen failure modes in composite materials. Delamination is often the main failure mode, which ultimately leads to final failure. (Trabal et al., 2022a) (Turon, 2007)

However, simply the presence of delamination does not necessarily mean the structure is

about to fail or at the end of its service life. (Turon, 2007). This is utilized in the damage tolerant design philosophy where the existence of delamination flaws in a structure are accepted as long as they are not critical.

Methods that can arrest or slow down delamination by increasing the through-thickness toughness in composite materials can therefore extend the service life of composite structures. This is beneficial from an economical point of view, especially when high performance structures are considered. Details on methods for improving toughness are explained later in section 1.3.

As mention above, delamination occurs due to loss of adhesion in a interface and it is caused by e.g. manufacturing defects, out of plane loading, intralaminar damage, abrupt changes in the layup, free edges, bonded joint, material configuration, temperature and moisture effects or a combination hereof (Trabal et al., 2022a) (Turon, 2007).

Looking at a microscopic scale, delamination growth follows in the wake of a damage zone formed at the delamination tip. The size, shape and evolution of this zone depends on e.g. the resin ductility, toughness and the state of loading. Further, once delamination growth occurs other mechanisms, such as fiber bridging and fiber breakage hinders further growth by increasing the fracture toughness (Turon, 2007).

Due to the high number of variables, delamination evolution is difficult to predict. However, such prediction are needed, in the damage tolerant design approach, as the degree of delamination is decisive for the structural integrity. Additionally, delamination can lead to a redistribution of the loads in a structure, thereby causing failure in other components (Turon, 2007). Prediction of delamination evolution requires advanced modelling techniques and in this work, delamination is modelled with cohesive zone modelling as explained in the next section.

1.2 Modelling of delamination

Delamination in composites is in essence a crack initiation and propagation problem. In continuum mechanics a crack, as illustrated in Figure 1.1, is defined as a discontinuous surface in a body. In its unloaded state the upper and lower crack surfaces are coincident and are only capable of transferring compression and shear stresses. (Bak, 2015)

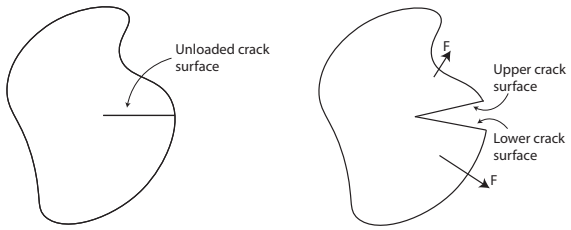


Figure 1.1. *Crack in a continuum.*

There exist three basic modes a crack can open in. These are mode I, mode II and mode III. Mode I is the mode that dissipates the smallest amount of energy. Therefore, regardless of the initial opening mode of the crack, it will gradually change direction during the crack propagation state until it reaches a mode I configuration. However, this is not necessarily true for laminated composite materials, as a crack in many cases is restricted to grow in the

interfaces in between the layers. Due to this, delamination cracks in composite materials can propagate in any mixed mode configuration. (Bak, 2015)

The modelling approaches of a crack growth problem, such as delamination in a laminate can broadly be divided into two modelling categories; linear elastic fracture mechanics (LEFM) and cohesive zone modelling (CZM). (Trabal et al., 2022b) In this work, the cohesive zone modelling approach is used.

The cohesive zone modelling approach is an attempt to accurately describe the crack, by introducing a cohesive zone. Contrary to the LEFM approach, where damage only develops at the crack tip, the introduction of the cohesive zone allows for representing material separation mechanisms e.g. fiber bridging, which extent quite far from the crack tip. This is beneficial when used to model composite materials, as these often have quite large damage zones. (Sun and Jin, 2012)

Another advantage of the cohesive zone modelling approach, compared to the LEFM modelling approach, is that it can be formulated such that both prediction of crack initiation and propagation are combined into one model. (Turon et al., 2006)

As mentioned above, the cohesive zone modelling approach introduces a cohesive zone. In the cohesive zone, tractions are added in order to balance the crack opening forces. By doing so, the stress singularity in front of the crack tip, inherent from LEFM, is removed (Bak, 2015). Thereby, the cohesive zone is normally described as cohesive surfaces held together by cohesive tractions, and failure is assumed to have occurred upon full separation of these surfaces. The size of this cohesive zone depends on the opening mode, where mode II and III generally have the largest zone (Turon, 2007).

The cohesive zone is placed either behind or in front of the crack tip, depending on the method used. However, in this work the crack tip is defined as the point, where the tractions are at a maximum and the cohesive zone is placed behind the crack tip as illustrated in Figure 1.2.

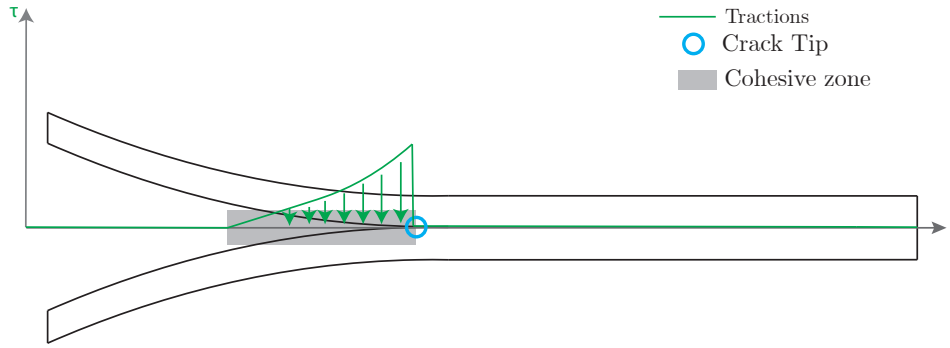


Figure 1.2. *Example of the cohesive zone with tractions in a DCB specimen.*

The relation between the cohesive tractions and the separation of the crack faces, which describes the material failure process, is formulated by a traction-separation law as stated in Equation 1.1. This is a generic expression that relates the separation, S, of the cohesive surfaces to the cohesive tractions, τ, in the cohesive zone.

$$\tau = \tau^o \, f(S/S^f) \tag{1.1}$$

Where τ^o and S^f in Equation 1.1 is used to describe critical values of the tractions and separation, respectively. τ^o is also referred to as onset traction, and defines damage initiation. S^f describes the separation distance, at which the interface strength is fully lost. f is a dimensionless function describing the shape of the traction-separation law. A generic traction-separation law is illustrated in Figure 1.3, and the area under this curve is defined as the critical energy release rate, G_c.

In order to apply the different cohesive traction-separation laws, it is necessary to determine cohesive zone related material properties, such as the onset traction, τ^o, critical energy release rate, G_c, or the critical separation distance, S^f (Sun and Jin, 2012).

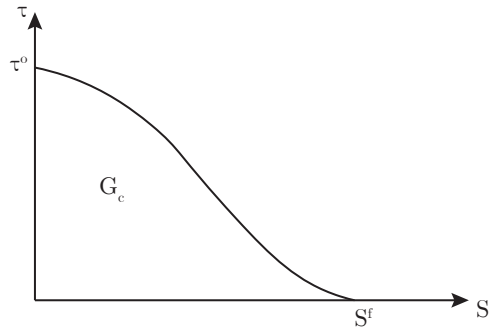


Figure 1.3. *Example of a generic cohesive traction-separation law. τ^o is the onset traction and S^f is the critical separation.*

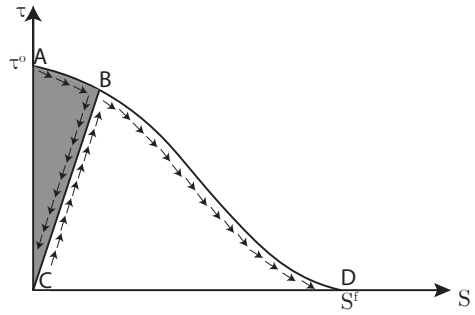


Figure 1.4. *Damage evolution during crack growth.*

In order to control the damage evolution in the model and ensure irreversible crack growth a damage parameter, d, can be introduced (Bak, 2015). This is explained with offset in Figure 1.4. Prior to loading there are no damage in the interface. If an overload is applied to the cracked body, i.e. the onset traction τ^o is exceeded, damage initiates. This results in the separation between the two cohesive surfaces gradually increases and the strength of the interface decreases. This is illustrated with curve A-B in Figure 1.4. During this, the value of the damage parameter, d, is increased according to the amount of interface degradation. (Bak, 2015)

If the overloading stage is followed by unloading, the traction separation law traces the linear curve B-C to the origin. In this stage the damage parameter, d, is constant as no further damage occurs. The energy dissipated during the overloading stage is equal to the area under the curve, marked with grey in Figure 1.4. If the crack is overloaded again, curve C-B is first traced without increasing the amount of damage and then curve B-D is followed and damage evolves (Bak, 2015).

In point, D, the separation equals the critical separation, S^f, and the tractions, τ, becomes zero, as illustrated in Figure 1.4. At this point the strength of the interface is lost and the amount of dissipated energy is equal to the critical energy release rate, G_c, for the interface. (Sun and Jin, 2012).

As mentioned, damage initiation is governed by the onset traction, τ^o, in Figure 1.3. Thus, if only one of the modes, either pure mode I, mode II, or mode III, are present in the crack, damage onset occurs when the interlaminar traction belonging to the mode, exceeds its onset traction τ^o.

Similar, crack propagation is governed by the energy release rate, G, and crack propagation starts when the energy release rate, G, equals the critical energy release rate, G_c. Thus, if only one of the modes are present in the crack, it starts to propagate as soon as the energy release rate equals the critical energy release rate for the specific mode i.e. the area under the traction-separation law. (Sun and Jin, 2012)

Under mixed mode conditions this is not as straight forward, and the traction-separation law must be able to take the mode interaction into account. This is necessary as the mode mixity e.g. affects how the crack propagates. (Bak, 2015) (Turon, 2007)

Cohesive zone modelling is often applied in the modelling of delamination in composite structures, as the interface between each layer in a laminate can be considered as a cohesive zone (Sun and Jin, 2012). This is normally done in a finite element framework, where there are generally two main methods for introducing the cohesive zone. This is done as either discrete inter-element cracks or discrete intra-element cracks, respectively (Turon 2007). In this work, the inter-element approach is used. In the inter-element approach cohesive elements are placed in the interfaces between the layers of the laminate, where the crack is expected to grow.

However, modelling crack initiation and propagation problems, with a cohesive zone modelling approach in a finite element framework is not unproblematic. Especially, the large variation of the tractions within and near the cohesive zone requires a fine element discretization. Trabal et al. (2022b) states that 3-10 elements must be used in the damage process zone, to get accurate results while also maintaining the computational efficiency. Additionally, a too coarse mesh may cause numerical oscillations in the structural response. This is caused by the inability of the coarse mesh to represent the tractions in the cohesive zone. This can further lead to convergence problems.

Thus, in a standard finite element program, modelling of crack propagation requires, that the full potential extension of the crack is modelled with equally fine elements. Due to the small length scale of the cohesive zone, relative to the full potential extension of the crack in large scale structures, the computational efficiency decreases (Trabal et al., 2022b).

1.3 Toughening of composites by promoting multiple delaminations

As mentioned in chapter 1, the damage tolerance of a structural component can be improved by using materials, which exhibits a toughening behaviour when cracks develops. Further, as explained in section 1.1, delamination is one of the main failure modes leading to final failure in composite structures and is typically caused by a weak interface between the individual layers.

In order to minimize the occurrence of delamination, several methods to toughen the interface are suggested in the literature. Some of the most common toughening methods are matrix modification, interleaving and through the thickness reinforcement.

These methods are normally introduced in interfaces that are highly exposed to delamination cracks. However, in this work a novel way of increasing the toughness is investigated.

Rask and Sørensen (2012) observed through experimental tests on DCB specimens, loaded with uneven bending moments, that onsetting new delamination in a composite material changes the damage mechanism which ultimately lead to a higher toughness. Specifically, their results showed, the steady-state fracture toughness increased in a almost proportional manner to the number of secondary cracks onset during the experiments. However, increasing the fracture toughness by onsetting new delamination cracks was not the focus of this article. Thus, it was only noted as a short remark without any further investigation.

Goutianos and Sørensen (2016) later showed, through numerical and analytical models, that the through-thickness fracture toughness in delamination problems can be increased, by onsetting multiple delaminations in the laminate. The analytical model developed in Goutianos and Sørensen (2016), is based on the J-integral and assumes pure mode I loading. This model predicts that the increase in the fracture toughness increases linearly with the number of new onset delamination cracks. Goutianos and Sørensen (2016) further verified through a numerical model, that the fracture toughness is increased by onsetting new delamination crack. However, the found fracture toughness in the numerical models did not increase linearly with the number of new delaminations as the toughness was found to also depend on the geometry and interface properties.

Trabal et al. (2022a) recently investigated the possibility for increasing the toughness, by promoting multiple delaminations by purposely inserting patches in specific locations in the interfaces. This was done through a numerical parameter study, where both weakening and toughening patches were introduced in double cantilever beam (DCB) specimens. The patches were modelled by cohesive interfaces with either increased or decreased interface properties, i.e. onset traction, τ^o and critical energy release rate, G_c. By introducing the patches Trabal et al. (2022a) were able to promote multiple delaminations for both cases, and as a consequence toughening the structure. The toughening effect is a result of an additional crack surface being formed, thereby dissipating more energy.

As mentioned traditional toughening methods is normally introduced in interfaces, that are highly exposed to delamination cracks. This often comes at a high cost, because the entire interface needs to covered. If instead the toughening methods is placed as patches, which promotes multiple delaminations the cost can be reduced (Trabal et al., 2022a). Furthermore using weakening patches might prove to be an even more cost effective way of promoting multiple delaminations(Trabal et al., 2022a).

The method of placing patches to promote delamination has only been studied numerically, and needs to verified through experimental work to confirm that the concept actually promote delamination and consequently toughens the structure.

Problem Statement 2

As explained in chapter 1, delamination is one of the main reasons for causing failure in composites. However, the presence of delamination in a structure does not necessarily mean that the structure is about to fail. This is exploited in the damage tolerant design approach, where advanced modelling techniques are used, along with non-destructive inspections, to extent the service life of a structure. Furthermore, the use of through-thickness toughening methods, which can arrest or slow down delamination propagation, by increasing the interface toughness, is important considerations within the damage tolerant design approach, when composite materials are considered.

In section 1.3 a new method for improving the structural toughness in composites was introduced. The work mentioned in section 1.3 has shown through analytical and numerical models, that the structural toughness during delamination propagation, can be improved by onsetting multiple delamination. According, to Trabal et al. (2022a) this toughening method has the potential for being a cost-effective alternative, to other commonly used toughening methods. It is also mentioned in section 1.3, that only limited experiments has been done on this subject to validate its applicability.

Thus, this thesis takes offset in the possibility for improving the structural toughness by onsetting multiple delaminations. This leads to the following problem statement.

> *"How can multiple delaminations in fiber composites be promoted by the use of toughening/weakening patches in order to improve damage tolerance in quasi-static loading?"*

2.1 Limitations and objectives

In order to answer the problem statement, this thesis is divided into two main parts. The first part includes chapter 3, 4 and 5 and focuses on the implementation of numerical solvers in a existing finite element framework. The objective of including this, is partly due to a learning goal set forth by the authors and as it is needed in order to model delamination problems in a finite element context.

The second part, of this thesis includes chapter 6, 7, 8, and 9. This part centers around experimental testing of DCB specimens to validate whether delamination can be promoted purposely by introducing either weakening or toughening patches in specific interfaces. In order to do this preliminary studies are firstly done to define a patch-placement which should promote new delamination cracks.

Only unidirectional glass fiber reinforced polymer (GFRP) composites with continuous fibers are experimentally tested in this work. Additionally, only DCB test specimens loaded in a mode I configuration are considered.

Finite Element Framework for Modelling Delamination

3

Delamination in composite materials is a complex phenomena and are therefore often modelled in a finite element framework. In this project, the finite element framework developed in Trabal et al. (2022b) and Trabal et al. (2022a), is used to model delamination in quasi static loaded 2D DCB-specimens. This chapter explains the main features of the finite element framework, which is hereafter referred to as the *MATLAB Framework*.

The MATLAB Framework is a FEA-code, which is specifically developed to simulate single or multiple delamination in double cantilever beam (DCB) specimens, end notched flexure (ENF) specimens, and mixed mode bending (MMB) specimens.

The MATLAB Framework is based on a Floating Node Method (FNM), and further includes additional novelties to improve on the highly computationally demanding process associated with modelling delamination with cohesive elements. In general, when modelling delamination with cohesive elements a very fine discretisation is needed near the crack tip, due to the presence of high gradient traction profiles. The reader is referred to section 1.2 for further details.

The main improvements in the MATLAB Framework, compared to already existing FNM includes a new adaptive FNM based element (A-FNM element). The A-FNM element is a general element that can use either solid or cohesive element formulation and refine and partition itself. The refinement of the A-FNM element is controlled with a novel Adaptive Refinement Scheme (ARS). In Figure 3.1 the basic principle of the MATLAB Framework is illustrated. The models initiates with a mesh consisting of a user defined number of A-FNM elements in the longitudinal direction and only one element through the thickness. The A-FNM elements are afterwards split and refined according to where delamination is occurring.

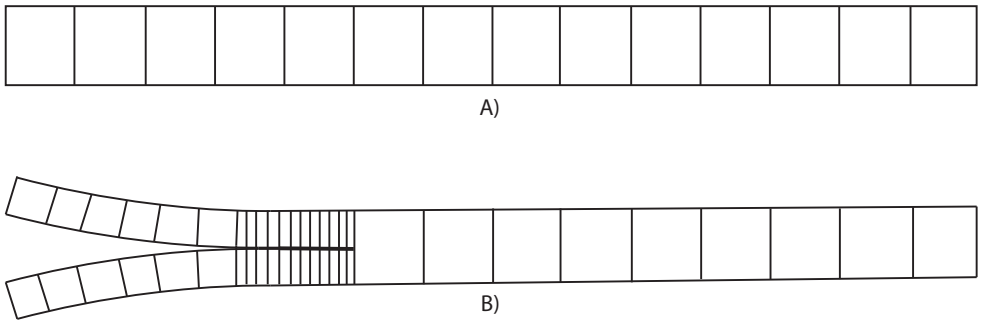


Figure 3.1. *A) The initial mesh consisting of non-refined A-FNM elements. B) A-FNM elements near the crack tip are refined to model the delamination crack.*

Both of these improvements are explained in detail later in this chapter. The improvements makes it possible to handle multiple simultaneous delaminations within each A-FNM element. Furthermore, it also enables automatic adaptment of the mesh refinement near the crack tip, according to changes in the damage processes zone, as the analysis evolves.

In the version of the MATLAB Framework used in this work it is possible to insert pre-cracks, weakening and toughening patches in any of the interfaces in the DCB specimen, as illustrated in Figure 3.2. The patches are modelled with decreased or increased critical energy release rate and onset traction compared to the virgin interfaces. Additionally, the MATLAB Framework is able to model delamination onset in pristine interfaces i.e. interfaces without any delamination.

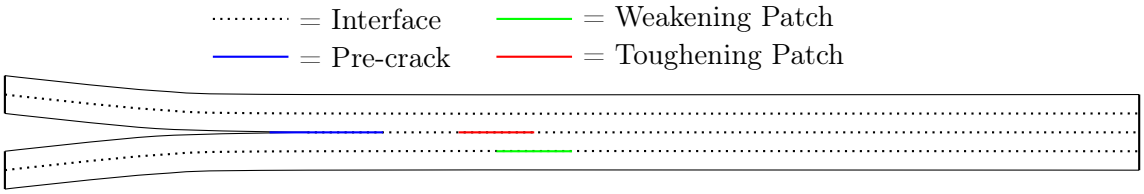


Figure 3.2. *Example of a DCB specimen with three interfaces, one pre-crack, one toughening patch and one weakening patch.*

The MATLAB-Framework consist of a pre-processor, a solver which in its original state only includes a force and displacement controlled Newton-Raphson solver and finally a post-processor. The first part of the preprocessing in the MATLAB Framework loads input files generated in ANSYS Mechanical-APDL. The input files defines the geometry, material properties, layup, boundary conditions, and mesh. This is followed by a preprocessing step in the MATLAB Framework that defines pre-cracks, patches, solver settings and A-FNM related variables.

The postprocessing step includes generation of figures and videos of traction profiles and the deformation of the specimen.

3.1 Floating Node Method and A-FNM Element

As mention, the MATLAB Framework is based on a Floating Node Method, where a number of floating nodes, in addition to the standard finite element nodes, are set up in the preprocessing part of the analysis. Each of the floating nodes are associated to a specific A-FNM element and are initially defined as inactive and only activated when needed. By activating or deactivated the floating nodes continuously, when and where needed, it is possible to dynamically refine and coarsen the A-FNM elements as well as dynamically add cohesive elements within the A-FNM element. By doing so, accuracy and computational efficiency is improved. Compability in the model is maintained by automatically generating multi-point-constraints (MPCs), when the A-FNM element refines or coarsens itself.

Thus, the main features of the novel A-FNM element is its ability to automatically refine, coarsen and split itself into 2D solid and interface elements during the analysis.

The principle of the A-FNM element is illustrated in Figure 3.3. Whenever a A-FNM element is refined, it activates some of the in-active floating nodes which are assigned

a position and displacement in the element by linear interpolation of the already active nodes. The newly activated floating nodes are used to split the A-FNM element into several sub-elements (SEs), and the relevant material is assigned to each of the sub-elements based on the A-FNM layup. When a floating node is activated, it is also included in the internal A-FNM element connectivity, that relates the floating nodes to the specific sub-element.

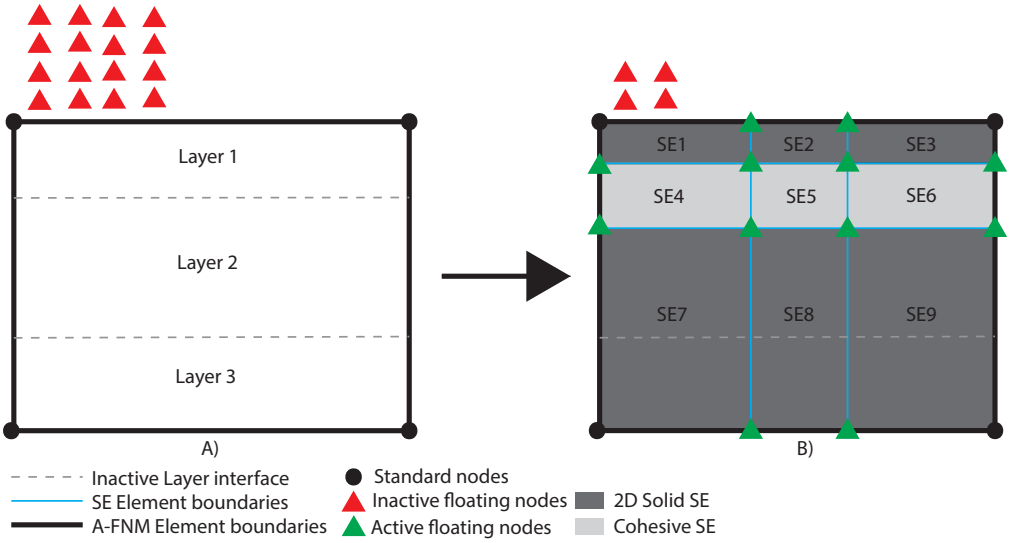


Figure 3.3. *Example of refinement of a A-FNM element. A) The A-FNM element is initialized as a standard 4-node element with inactive floating nodes. B) Some of the floating nodes are activated in order to refine and split the A-FNM element. The element is split with cohesive elements in the interface between layer 1 and layer 2. This further requires the A-FNM element to be refined with solid sub-elements.*

The A-FNM is able to refine itself both through the thickness of the element and in the longitudinal direction of the element. The refinement through thickness is dependant on the number of interfaces, which needs to be modelled with cohesive elements. The refinement in the longitudinal direction is done according to a user defined value in the MATLAB Framework, that specifies the number of cohesive sub-elements, that are to be used within each refined A-FNM element.

An advantages of using a FNM is the possibility to utilise a standard finite element formulation, where all operations are done on element level. This furthermore makes it possible to keep the standard finite element bookkeeping procedures. Thereby, the efficiency of the calculations are improved compared to other re-meshing techniques.

Thus, standard finite element procedures are used within each sub-element, to calculate the tangent stiffness matrix and internal force vector at sub-element level. These are afterwards assembled to the A-FNM element level. In summary, if the element needs to be split in an interface, refined or both, the following four steps are done:

1. Calculate divisions in element
2. Activate floating nodes
3. Calculate Subelement tangent stiffness matrix and internal force vector
4. Apply MPCs if needed.

If there is no need for any refinement of the A-FNM element, the tangent stiffness and internal force vector is calculated only at A-FNM level.

As mentioned, the A-FNM element uses two different element formulations. These are a 2D n-layered, 4-noded, geometrically linear, solid, plane strain element with the option of using enhanced assumed strains stabilization and a 4-noded zero-thickness cohesive element used to model the cohesive zone. The enhanced assumed strains stabilization scheme is only used for the coarse 2D solid-elements and are deactivated in the refined elements. Each A-FNM element is initialized as a 4-noded solid element, which can be split, as explained above. The cohesive element used is based on Turon et al. (2006) and Camanho et al. (2003), which utilizes a progressive interlaminar damage formulation, based on a bilinear cohesive law and accounts for geometrically non-linear effects. The reader is referred to Appendix A for further details.

3.2 Adaptive Refinement Scheme for Multiple Cracks

As mentioned, the purpose of the Adaptive Refinement Scheme (ARS) is control the level of refinement in each A-FNM element. This is needed to ensure an ideal discretization in all interfaces, which ensures accurate modelling without reducing computational efficiency. The discretization is checked and updated for each iteration in the solver, based on the damage state and the traction profile in the interface.

The traction profile can be divided into four regions which need different discretisation, in order to model the traction profile accurately while maintaining the computational efficiency. This is done by assigning the A-FNM elements within each region a corresponding refinement state. However the main driver for the ARS is the damage parameter, d, since it is easily accessible from the cohesive elements and related to the traction profile illustrated in Figure 3.4. d is used in the cohesive element formulation and defines the degree of damage in each cohesive element. In the ARS the four regions are related to the damage parameter in the following way:

- **Full damage region:** The full damage region is defined as the region where the interfacial stiffness is lost. The damage parameter in this region is equal to 1. The A-FNM elements in this zone is assigned a full damage state and the interface is meshed with coarse cohesive elements, with the single purpose of preventing interpenetration of the crack faces.
- **High gradient region:** The high gradient region is defined as the region where the gradient of the tractions is high. The high gradient region includes the damage process zone and a small distance ahead of it. The damage process zone is defined by the damage parameter being in the interval $0 \leq d \leq 1$. Due to the high gradients, the A-FNM elements in this region is assigned a refined state and the entire element including the cohesive elements in the interface is refined. Generally, 3-10 cohesive elements are needed in the damage process zone in order to achieve accurate results and fast convergence.
- **Low gradient region:** The low gradient region is placed in front of the high gradient region and has a damage parameter, d, equal to zero. The A-FNM elements in this region is assigned a coarse state and the interface is modelled with coarse cohesive elements. The low gradient region needs to be meshed with coarse cohesive elements in order to determine the stored elastic energy, as this is important to achieve smooth and accurate results. As the gradient of the traction profile is low in this region,

it can be modelled with coarse cohesive elements without affecting the accuracy significantly.

- **Inactive region:** The inactive region is defined by having no or a very low gradients in the traction profile present. The damage parameter, d in this region is equal to zero. The elements in this zone is assigned an inactive state and does not have to be discretised with cohesive elements.

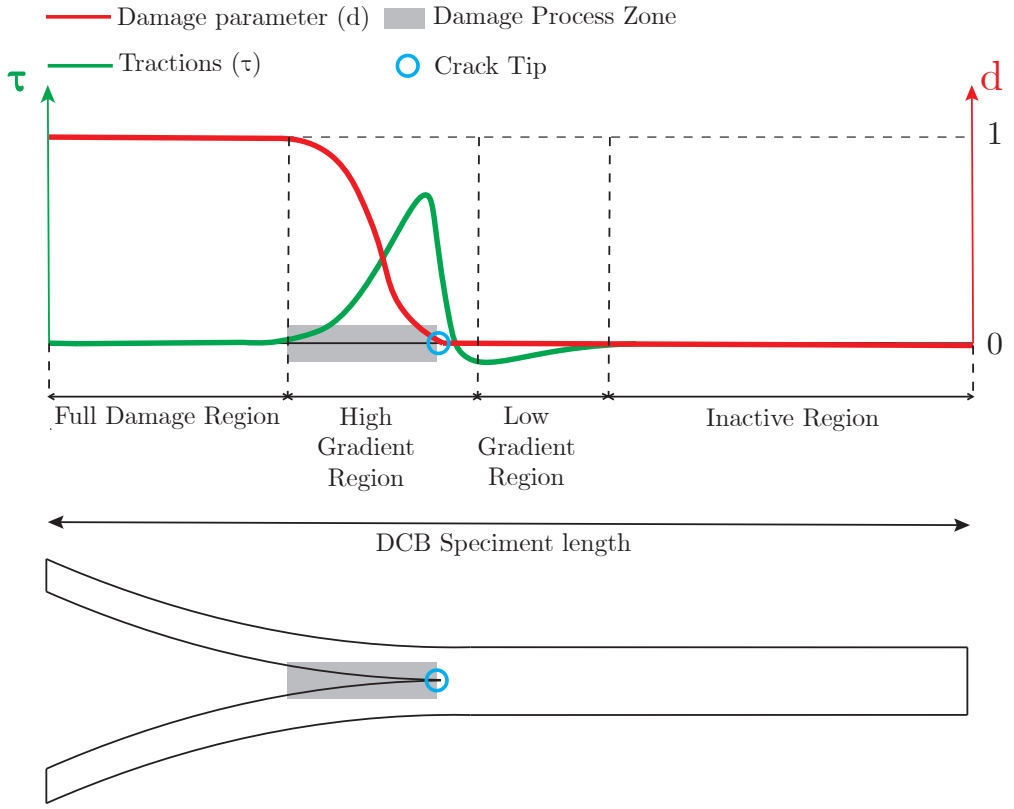


Figure 3.4. *Illustration of the four different regions and associated damage parameter, d.*

If multiple delaminations occur in the same A-FNM element, the interface requiring the most refined state becomes dominant for the refinement of the A-FNM element.

Thereby each interface in the model is assigned either a full damage state, a refined state, a coarse state or inactive state depending on which of the four regions it is located in.

The interface state is decisive for the mesh refinement of the A-FNM element

3.2.1 ARS-Related Variables

There are five user-inputtet ARS-related variables in the MATLAB Framework, that are used to control the ARS. The ARS-related variables are listed and explained below:

- $\mathbf{R_{Zone1}}$: Specifies the refinement used in the full damage state A-FNM elements, i.e. where $d = 1$. Thus, this variable controls the longitudinal refinement of the model in zone 1 in Figure 3.5. The refinement is specified as an integer number, and defines how many times the full damage state A-FNM elements are split in the longitudinal direction. This is illustrated in Figure 3.5 B), where R_{Zone1} is equal to two, which means the length of the sub-elements, are half of the length of the A-FNM elements.

- **R_{Zone23}**: Specifies the refinement used in the refined state A-FNM elements. Thus, this variable controls the longitudinal refinement in region 2 and 3 in Figure 3.5. Zone 2 is where damage is occurring, i.e. $0 < d < 1$. The refinement in zone 2 and 3 is specified as an integer number, which defines the number of sub-elements that are to be used in the longitudinal direction within each refined state A-FNM element. This is illustrated in Figure 3.5 C), where R_{Zone23} is set equal to 3.
- **N_{Zone3}**: Specifies the number of A-FNM elements in front of the crack tip which are assigned a refined state. Thus, this variable defines how many A-FNM elements are included in zone 3 in Figure 3.5 A). As mentioned region 3 is refined in the same way as zone 2.
- **N_{Zone4}**: Specifies the number of A-FNM elements ahead of region 3, which are split and meshed with cohesive zone elements in the interface. Thus, this variable defines how many A-FNM elements that are included in zone 4 in Figure 3.5.
- **R_{Zone4}**: Specifies the refinement of the coarse state A-FNM elements in zone 4 in Figure 3.5 A). This is specified as an integer number, and defines how many times the A-FNM elements are split in the longitudinal direction, equivalent to R_{Zone1}.

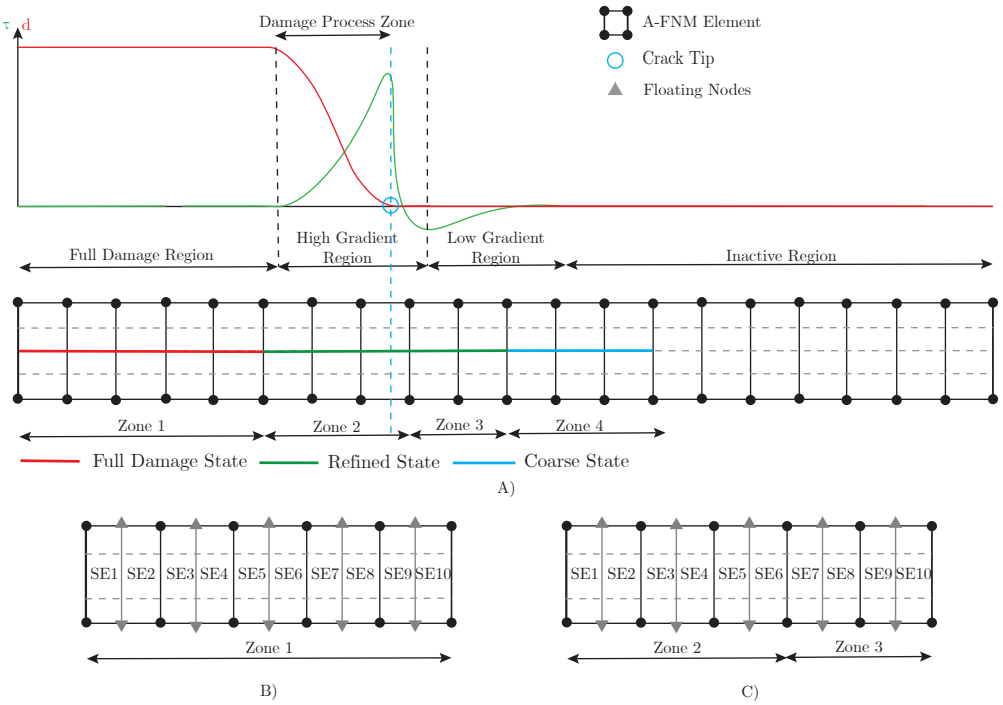


Figure 3.5. *Refinement principle based on the user defined ARS variables for a DCB specimen with one interface.*

In the models done in this project the entire interface is as a minimum discretised by use of the coarse state, by setting N_{Zone3} equal to the number of A-FNM elements in the model. This is done, as non-physical phenomena may be encountered, in the transition from the coarse state to the idle state, due to the MPCs. Furthermore, variable R_{Zone1} and R_{Zone4} are kept at a constant value of one, meaning no further refinement is done in these zones.

Geometric Arc Length Solver 4

Crack propagation problems, and thereby also the delamination process which might occur in composite materials, are non-linear problems (Jensen, 2021). In its original form, the MATLAB Framework presented in chapter 3 is not able to fully solve this inherent non-linear behaviour, due to the solver used. In the following chapter a brief outline of the general governing equations and solution procedures related to non-linear problems is presented. This is followed by a short explanation of the issues encountered when using the existing solver in the MATLAB Framework. Finally, to circumvent the issues encountered when solving delamination problems with non-linear behaviour, the arc-length method is presented and implemented in the MATLAB-Framework.

In this chapter, curly brackets, $\{*\}$, are used to indicate vectors, square brackets, $[*]$, are used to indicate matrices, and symbols without brackets indicate scalar values.

In general, a non-linear solution algorithm takes offset in the equilibrium equations written in residual form, as done in Equation 4.1.

$$\big\{\boldsymbol{R}\big(\{\boldsymbol{D}\},\{\boldsymbol{F_{ext}}\}\big)\big\} = \{\boldsymbol{F}_{int}(\{\boldsymbol{D}\})\} - \{\boldsymbol{F}_{ext}\} = \{\boldsymbol{0}\} \tag{4.1}$$

where $\{\boldsymbol{F}_{ext}\}$ is the external forces, $\{\boldsymbol{F}_{int}(\{\boldsymbol{D}\})\}$ is the internal forces in the system which depends on the displacements $\{\boldsymbol{D}\}$. $\big\{\boldsymbol{R}\big(\{\boldsymbol{D}\},\{\boldsymbol{F_{ext}}\}\big)\big\}$ is the residual. The residual is introduced as the internal forces and external forces are generally not in equilibrium for an arbitrary displacement $\{\boldsymbol{D}\}$ and an external load $\{\boldsymbol{F}_{ext}\}$. The residual is as such used to find corrections for the displacements fields, in order to achieve equilibrium in the structure (Vasios, 2015).

For a non-linear finite element model, Equation 4.1 is often solved by an incremental-iterative approach, where the prescribed loads on the structure is applied in several substeps, and where the iterative method is used to ensure the residual force is sufficiently small within each substep. Using such a incremental-iterative approach for a force controlled problem with proportional loading, the equilibrium equation for the *n'th* substep is usually defined by Equation 4.2.

$$\big\{\boldsymbol{R}\big(\{\boldsymbol{D}\},\{\boldsymbol{F}_{ext}\}\big)\big\} = \{\boldsymbol{F}_{int}(\{\boldsymbol{D}\})\}^n - \lambda^n\{\hat{\boldsymbol{F}}_{ext}\} = \{\boldsymbol{0}\} \tag{4.2}$$

Where $\{\boldsymbol{F}_{int}(\{\boldsymbol{D}\})\}^n$ is the internal force in the *n'th* substep. λ^n is a scalar-valued load-level parameter for the *n'th* substep, which is multiplied with the normalised external load vector $\{\hat{\boldsymbol{F}}_{ext}\}$. By introducing the scalar-valued load-level parameter, it is possible to apply the external load in several substeps (de Borst et al., 2012).

Often iterative methods such as Newton-Raphson's method is utilized as is the case in the MATLAB Framework. In this method the non-linear governing equations, in Equation 4.2,

are linearized using truncated Taylor series and solved iteratively, in order to achieve convergence between the internal and external applied loads. Thereby, in a force controlled setting, the method works by increasing the load parameter by a predefined value, $\Delta\lambda$, for each new substep and calculating the incremental displacement field, $\{\Delta \boldsymbol{D}\}$, which ensures equilibrium (Vasios, 2015).

Defining the previous converged substep by $(\{\boldsymbol{D}\}^{n-1}, \lambda^{n-1})$, and assuming that an increment in both load-level parameter $\Delta\lambda$ and displacement $\{\Delta \boldsymbol{D}\}$ has been found, that ensures equilibrium, then λ^{n-1} and $\{\boldsymbol{D}\}^{n-1}$ are updated according to Equation 4.3 and 4.4, to yield the next point on the equlibrium curve $(\{\boldsymbol{D}\}^{n}, \lambda^{n})$.

$$\lambda^n = \lambda^{n-1} + \Delta\lambda \tag{4.3}$$

$$\{\boldsymbol{D}\}^n = \{\boldsymbol{D}\}^{n-1} + \{\Delta \boldsymbol{D}\} \tag{4.4}$$

However, Newton-Raphson's method as explained above, has some limitation and will fail to converge or skip part of the curve, if the equilibrium curve exhibits snap-through behaviour, i.e. it has a limit point as illustrated in Figure 4.1. This is caused by the tangent stiffness becoming singular in these points and due to the requirement that the load-parameter λ must increase for each substep. This problem can in some cases be circumvented by utilizing a displacement controlled solution method. However, a displacement controlled solution encounters the same problems if the structure has a snap-back behaviour, i.e. it has a turning point as illustrated in Figure 4.1. The dotted lines in Figure 4.1 are used to indicate that part of the equilibrium curve may be skipped. (Vasios, 2015)

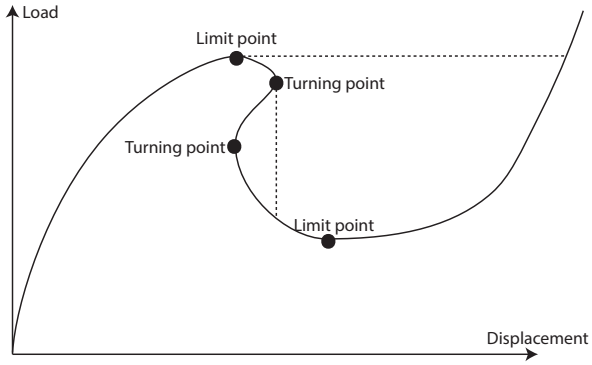


Figure 4.1. *Limitations of the force controlled and displacement controlled Newton-Raphson solvers.*

Because the equilibrium curve is generally unknown, it is not possible to determine beforehand, whether the structure exhibits snap-back or snap-through behaviour and thereby, whether Newton's method can be applied or not. Alternatively, more advanced solution algorithms, such as the arc-length solver can be utilized.

The rest of the chapter is organised in the following way; first, an arc-length method based on a geometric constraint is presented, implemented in the MATLAB Framework and validated. This is followed by a presentation, implementation and validation of two arc-length methods that use energy dissipation based constraints. These constraint are beneficial when solving problems with local material instability such as delamination problems.

4.1 Geometric Arc-Length Method

The arc-length method, presented in this section is based on Crisfield (2000) and de Borst et al. (2012). The arc-length method takes offset in the residual equilibrium equation defined in Equation 4.2. However, in the arc-length method, both the displacement $\{\boldsymbol{D}\}$, and load-level parameter, λ, are variables in the iterative solution scheme. This is different from other commonly used solvers, such as the Newton-Raphson solver explained above.

The basic idea of the arc-length method is thereby to find the next equilibrium point on the load-displacement curve as the intersection of the load-displacement curve and a surface defined by a *path-following constraint*. This is illustrated for a 1 degree of freedom system in Figure 4.2, where the surface defined by the path-following constraint is a circle. Also illustrated in Figure 4.2, is all the iterations, within the illustrated substep until convergence is achieved. From the figure it is evident, that all these iterations lies on the path-following constraint.

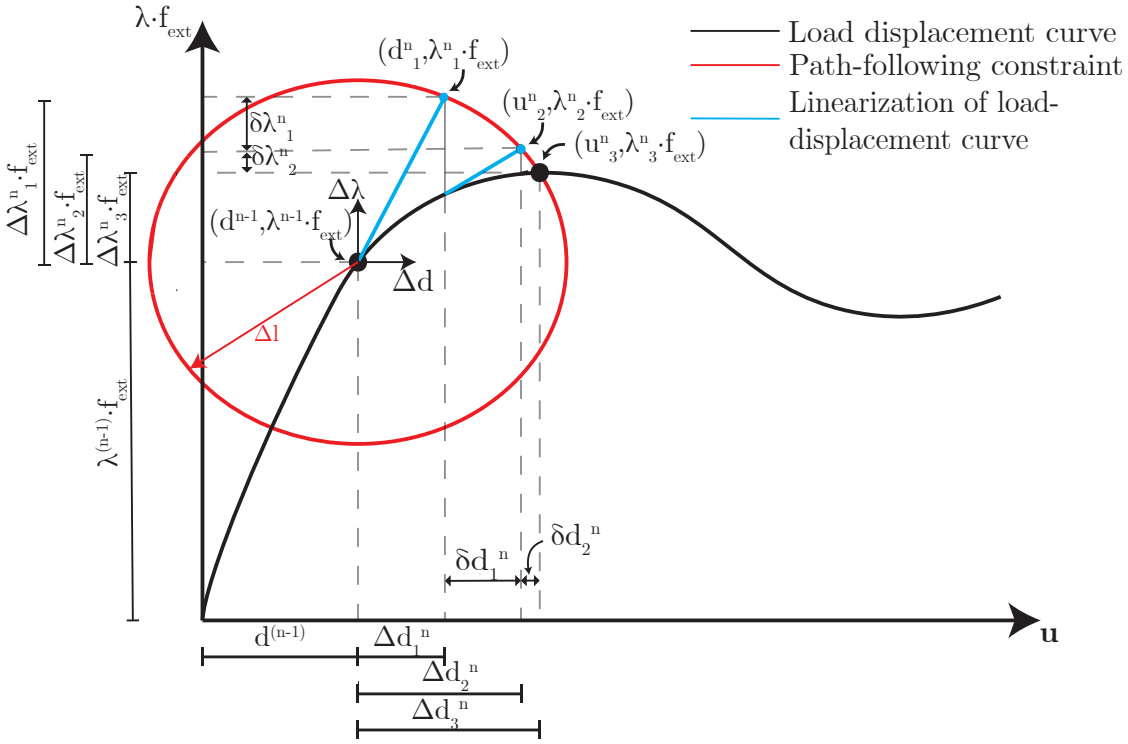


Figure 4.2. *One degree of freedom illustration of the arc-length solver.*

In the following, superscript, n, is used to define the current substep and subscript, i, is used to define iteration number in the substep. Furthermore, D_i^n and λ_i^n are used to describe the displacement and load-level parameter. ΔD_i^n and $\Delta\lambda_i^n$ are used to describe the incremental displacement field and load-level parameter. δD_i^n and $\delta\lambda_i^n$ are used to describe corrections to the displacement field and load-level parameter from one iteration to the next, within the substep.

Taking offset in a previous converged substep $(d^{n-1}, \lambda^{n-1} f_{ext})$, in Figure 4.2, the method initiate each substep by calculating a predictor step, which leads to the point $(d_1^n, \lambda_1^n f_{ext})$ by varying the load-level parameter and displacement by $\Delta\lambda_1$ and Δd_1, respectively. This predictor step is explained in details in subsection 4.1.1. Based on this new point,

$(d_1^n,\lambda_1^n f_{ext})$, the residual is determined. If the residual is too large, the iterative process of the arc-length solver algorithm continues, and it finds the next point on the path-following constraint $(d_2^n,\lambda_2^n f_{ext})$ by adding corrections, δd_1, and $\delta\lambda_1$. If the residual is still to large the algorithm proceeds in a similar manner and finds the next points on the path-following constraint, $(d_3^n,\lambda_3^n f_{ext})$, $(d_4^n,\lambda_4^n f_{ext})\dots(d_i^n,\lambda_i^n f_{ext})$ iteratively. This is done until the intersection of the load-displacement curve and the path-following constraint is found which implies the current substep has converged and a new point on the load-displacement curve is determined.

In the arc-length method the load-level parameter λ in Equation 4.2 is included as an additional unknown variable in the system of equations. This differs from the Newton-Raphson method where it is a prescribed incrementally increasing parameter. Thus, the original problem with m unknown displacements $\{\boldsymbol{D}\}$ is increased to a problem containing $m+1$ unknowns. In order to solve this system of equations, it is necessary to add one additional equation, which is done through the path-following constraint,which is defined in a general form in Equation 4.5.

$$g(\{\boldsymbol{D}\},\lambda,\{\Delta\boldsymbol{D}\},\Delta\lambda,\Delta l)=0 \tag{4.5}$$

Where Δl defines the distance between the previous converged substep to the path-following constraint in the load-displacement space. Thereby, Δl, as illustrated in Figure 4.2, determines the size of the incremental step in the $m+1$ load-displacement space. The augmented system of equations can be expressed as done in Equation 4.6.

$$\begin{Bmatrix} \{\boldsymbol{F}_{int}(\{\boldsymbol{D}\})\}^n-\lambda^n\{\boldsymbol{F}_{ext}\} \\ g^n \end{Bmatrix}=\begin{Bmatrix}\{\boldsymbol{0}\}\\0\end{Bmatrix} \tag{4.6}$$

The system of non-linear equations can be solved iteratively with a Newton-Raphson approach, where the equations are linearsed by a truncated Taylor series expansion, as done in Equation 4.7.

$$\begin{Bmatrix} \{\boldsymbol{F}_{int}\}_i^n-\lambda_i^n\{\hat{\boldsymbol{F}}_{ext}\}+\frac{\partial\{\boldsymbol{F}_{int}\}}{\partial\{\boldsymbol{D}\}}\{\delta\boldsymbol{D}\}_i^n-\{\hat{\boldsymbol{F}}_{ext}\}\delta\lambda_i^n \\ g_i^n+\left(\frac{\partial g}{\partial\{\boldsymbol{D}\}}\right)^T\{\delta\boldsymbol{D}\}_i^n+\frac{\partial g}{\partial\lambda}\delta\lambda_i^n \end{Bmatrix}=\begin{Bmatrix}\{\boldsymbol{0}\}\\0\end{Bmatrix} \tag{4.7}$$

By rewriting Equation 4.7 it is possible to solve for corrections, $\{\delta\boldsymbol{D}\}$ and $\delta\lambda$ to the displacement field and load-level parameter, as done in Equation 4.8.

$$\begin{bmatrix}[\boldsymbol{K}_t]_i^n & -\{\hat{\boldsymbol{F}}_{ext}\}\\ (\{\boldsymbol{h}\}_i^n)^T & w_i^n\end{bmatrix}\begin{Bmatrix}\{\delta\boldsymbol{D}\}_i^n\\ \delta\lambda_i^n\end{Bmatrix}=\begin{Bmatrix}\{\boldsymbol{R}\}_i^n\\ -g_i^n\end{Bmatrix} \tag{4.8}$$

Where the tangent stiffness, $[\boldsymbol{K}_t]$, a h-vector, $\{\boldsymbol{h}\}$, and scalar, w, is introduced and defined in Equation 4.9.

$$[\boldsymbol{K}_t]_i^n=\frac{\partial\{\boldsymbol{F}_{int}\}}{\partial\{\boldsymbol{D}\}}\qquad\{\boldsymbol{h}\}_i^n=\frac{\partial g}{\partial\{\boldsymbol{D}\}}\qquad w_i^n=\frac{\partial g}{\partial\lambda} \tag{4.9}$$

The arc-length method as introduced in Equation 4.8 and 4.9 is known as Riks arc-length method. A detailed explanation of Riks arc-length method and its implementation in the MATLAB Framework is presented in Appendix B.

Riks arc-length method has the disadvantage that the matrix in Equation 4.8 destroys the symmetry and banded structure of the tangential stiffness matrix according to de Borst et al. (2012). For this reason, more efficient arc-length algorithms have been proposed and the following takes offset in the method by Crisfield (2000).

Taking offset in the first of the equations in Equation 4.7, introducing the tangent stiffness $[\boldsymbol{K}_t]$ as in Equation 4.9 and rewriting yields Equation 4.10.

$$[\boldsymbol{K}_t]_i^n\{\delta \boldsymbol{D}\}_i^n = -(\{\boldsymbol{F}_{int}\}_i^n - \lambda_i^n\{\hat{\boldsymbol{F}}_{ext}\} - \delta\lambda_i^n\{\hat{\boldsymbol{F}}_{ext}\}) \tag{4.10}$$

Introducing the residual force as $\{\boldsymbol{R}\}_i^n = \{\boldsymbol{F}_{int}\}_i^n - \lambda_i^n\{\hat{\boldsymbol{F}}_{ext}\}$ yields Equation 4.11.

$$[\boldsymbol{K}_t]_i^n\{\delta \boldsymbol{D}\}_i^n = -\big(\{\boldsymbol{R}\}_i^n - \delta\lambda_i^n\{\hat{\boldsymbol{F}}_{ext}\}\big) \tag{4.11}$$

Isolating for the displacement correction, $\{\delta \boldsymbol{D}\}$, and assuming the tangent stiffness, $[\boldsymbol{K}_t]$, to be non-singular yields Equation 4.12.

$$\{\delta \boldsymbol{D}\}_i^n = -([\boldsymbol{K}_t]_i^n)^{-1}\{\boldsymbol{R}\}_i^n + \delta\lambda_i^n([\boldsymbol{K}_t]_i^n)^{-1}\{\hat{\boldsymbol{F}}_{ext}\} = \{\delta \boldsymbol{D}^{II}\}_i^n + \delta\lambda_i^n\{\delta \boldsymbol{D}^{I}\}_i^n \tag{4.12}$$

where:

$$\{\delta \boldsymbol{D}^{II}\}_i^n = ([\boldsymbol{K}_t]_i^n)^{-1}\{\boldsymbol{R}\}_i^n \qquad \text{and} \qquad \{\delta \boldsymbol{D}^{I}\}_i^n = ([\boldsymbol{K}_t]_i^n)^{-1}\{\hat{\boldsymbol{F}}_{ext}\} \tag{4.13}$$

In this way the iterative displacement correction $\{\delta \boldsymbol{D}\}_i^n$ is split into two parts. $\{\delta \boldsymbol{D}^{II}\}$ is the iterative correction in displacement due to residual forces, and equivalent to the step used in the Newton-Raphson method, and $\{\delta \boldsymbol{D}^{I}\}$ is the iterative correction in displacement due to the external load vector. Common for both parts of the displacement corrections $\{\delta \boldsymbol{D}^{I}\}$ and $\{\delta \boldsymbol{D}^{II}\}$ are that they only rely on known quantities and can therefore be calculated.

In Equation 4.12 and 4.13 it is assumed that the tangent stiffness $[\boldsymbol{K}_t]$ is always non-singular, which is the case for all points along the equilibrium curve except at limit points. However, according to Crisfield (2000) it is very unlikely, that the solution ends up exactly at a limit point, which makes the tangent stiffness singular. For this reason the non-singular requirement for the tangent stiffness $[\boldsymbol{K}_t]$ does not in practice invalid the method.

Based on the computed displacement correction in Equation 4.12, the new incremental displacement in the current substep is updated by Equation 4.14.

$$\{\Delta \boldsymbol{D}\}_{i+1}^n = \{\Delta \boldsymbol{D}\}_i^n + \{\delta \boldsymbol{D}\}_i^n = \{\Delta \boldsymbol{D}\}_i^n + \big(\{\delta \boldsymbol{D}^{II}\}_i^n + \delta\lambda_i^n\{\delta \boldsymbol{D}^{I}\}_i^n\big) \tag{4.14}$$

Equivalently, the new incremental load-level parameter in the current substep is updated by Equation 4.15.

$$\Delta\lambda_{i+1}^n = \Delta\lambda_i^n + \delta\lambda_i^n \tag{4.15}$$

However, $\delta\lambda_i^n$ in Equation 4.14 and 4.15 is still unknown and must be determined. This is done by using a path-following constraint. In the following the widely used path-following constraint defined below is utilized.

$$g = \big(\{\Delta \boldsymbol{D}\}_i^n\big)^T\{\Delta \boldsymbol{D}\}_i^n + \beta^2(\Delta\lambda_i^n)^2\{\hat{\boldsymbol{F}}_{ext}\}^T\{\hat{\boldsymbol{F}}_{ext}\} - \Delta l^2 = 0 \tag{4.16}$$

β is a user-defined variable, which can be changed to achieve different path-following constrains. Two commonly used path-following constrains are the spherical path-following

constrain, $\beta = 1$, and the cylindrical path-following constrain, $\beta = 0$. According to de Borst et al. (2012) the value of β does not change the performance of the algorithm much. de Borst et al. (2012) also concludes that $\beta = 0$ yields a robust method for most engineering problems.

As the incremental step size, Δl, illustrated in Figure 4.2, is kept constant for all substeps, it is possible to write Equation 4.17

$$\begin{aligned}(\{\Delta \boldsymbol{D}\}_i^n)^T\{\Delta \boldsymbol{D}\}_i^n + \beta^2(\Delta\lambda_i^n)^2\{\hat{\boldsymbol{F}}_{ext}\}^T\{\hat{\boldsymbol{F}}_{ext}\} \\ = (\{\Delta \boldsymbol{D}\}_{i+1}^n)^T\{\Delta \boldsymbol{D}\}_{i+1}^n + \beta^2(\Delta\lambda_{i+1}^n)^2\{\hat{\boldsymbol{F}}_{ext}\}^T\{\hat{\boldsymbol{F}}_{ext}\} = \Delta l^2\end{aligned} \tag{4.17}$$

Substituting the expression for $\{\Delta \boldsymbol{D}_{i+1}\}$ from Equation 4.14 and the expression for $\Delta\lambda_{i+1}$ from 4.15 into Equation 4.17 yields Equation 4.18.

$$\begin{aligned}g = \left[\{\Delta \boldsymbol{D}\}_i^n + \{\delta \boldsymbol{D}^{II}\}_i^n + \delta\lambda_i^n\{\delta \boldsymbol{D}^{I}\}_i^n\right]^T \left[\{\Delta \boldsymbol{D}\}_i^n + \{\delta \boldsymbol{D}^{II}\}_i^n + \delta\lambda_i^n\{\delta \boldsymbol{D}^{I}\}_i^n\right] \\ +\beta^2(\Delta\lambda_i^n + \delta\lambda_i^n)^2\{\hat{\boldsymbol{F}}_{ext}^T\}\{\hat{\boldsymbol{F}}_{ext}\} - \Delta l^2 = 0\end{aligned} \tag{4.18}$$

Rewriting this equations yields the quadratic equation defined in Equation 4.19.

$$g = a_1(\delta\lambda_i^n)^2 + a_2\delta\lambda_i^n + a_3 = 0 \tag{4.19}$$

where:

$$\begin{aligned}a_1 &= \left(\{\delta \boldsymbol{D}^I\}_i^n\right)^T\{\delta \boldsymbol{D}^I\}_i^n + \beta^2\{\hat{\boldsymbol{F}}_{ext}\}^T\{\hat{\boldsymbol{F}}_{ext}\} \\ a_2 &= 2\left[\left(\left(\{\Delta \boldsymbol{D}\}_i^n\right)^T + \left(\{\delta \boldsymbol{D}^{II}\}_i^n\right)^T\right)\{\delta \boldsymbol{D}^I\}_i^n + \beta^2\Delta\lambda_i^n\{\hat{\boldsymbol{F}}_{ext}\}^T\{\hat{\boldsymbol{F}}_{ext}\}\right] \\ a_3 &= (\{\Delta \boldsymbol{D}\}_i^n + \{\delta \boldsymbol{D}^{II}\}_i^n)^T(\{\Delta \boldsymbol{D}\}_i^n + \{\delta \boldsymbol{D}^{II}\})_i^n + \beta^2(\Delta\lambda_i^n)^2\{\hat{\boldsymbol{F}}_{ext}\}^T\{\hat{\boldsymbol{F}}_{ext}\} - \Delta l^2\end{aligned} \tag{4.20}$$

By solving the quadratic equation it is possible to determine $\delta\lambda_i^n$. Having determined this value, $\{\Delta \boldsymbol{D}\}_{i+1}^n$ and $\Delta\lambda_{i+1}^n$ can be determined from Equation 4.14 and Equation 4.15, respectively.

However, as Equation 4.19 is a quadratic equation it generally has two solutions. The correct solution must be chosen each time, such that the method evolves forward. An often chosen approach is to calculate which incremental displacement vector, $\{\Delta \boldsymbol{D}_{i+1}^n\}$, that points in the same direction as the previous incremental displacement vector $\{\Delta \boldsymbol{D}_i^n\}$. This can be done by utilizing the definition of the dot-product, where the root which yields the smallest angle and thereby the largest cosine value is chosen, as done in Equation 4.21. Alternatively, Equation 4.16 can be linearised which means only one value for $\delta\lambda$ is found.

$$\cos(\theta) = \frac{(\{\Delta \boldsymbol{D}\}_i^n)^T\{\Delta \boldsymbol{D}_{i+1}^n\}}{|\{\Delta \boldsymbol{D}\}_i^n| \cdot |\{\Delta \boldsymbol{D}_{i+1}^n\}|} \tag{4.21}$$

An additional problem is if the solution to Equation 4.19 yields imaginary roots. If this happens, a often used approach is to restart the solution algorithm, from the last converged substep and reduce the incremental step Δl.

Once the correct $\delta\lambda_i$ is determined the displacement and load level parameter is updated according to Equation 4.14 and 4.15.

4.1.1 Predictor Solution

In order to start the iterative solution algorithm in each substep, a predictor step is needed for the first iteration. In this predictor step the displacement increment is based on a forward-Euler tangential predictor and defined in Equation 4.22.

$$\{\Delta \boldsymbol{D}\}_1^n = \Delta\lambda_1^n([\boldsymbol{K}_t]_1^n)^{-1}\{\boldsymbol{F}_{ext}\} \tag{4.22}$$

This is in accordance with Equation 4.12, if it is assumed that the residual in the previous converged substep is zero. Inserting this predictor term in the path-following constraint defined in 4.16 yields Equation 4.23.

$$\left(\Delta\lambda_1^n([\boldsymbol{K}_t]_1^n))^{-1}\{\hat{\boldsymbol{F}}_{ext}\}\right)^T\left(\Delta\lambda_1^n([\boldsymbol{K}_t]_1^n)^{-1}\{\hat{\boldsymbol{F}}_{ext}\}\right) + \beta^2\lambda_1^n\{\hat{\boldsymbol{F}}_{ext}\}^T\{\boldsymbol{F}_{ext}\} - \Delta l^2 = 0 \tag{4.23}$$

Which can be rewritten to Equation 4.24.

$$(\Delta\lambda_1^n)^2 = \frac{\Delta l^2}{\left(((([\boldsymbol{K}_t]_1^n)^{-1}\{\hat{\boldsymbol{F}}_{ext}\})^T(([\boldsymbol{K}_t]_1^n)^{-1}\{\hat{\boldsymbol{F}}_{ext}\}) + \beta^2\{\hat{\boldsymbol{F}}_{ext}\}^T\{\boldsymbol{F}_{ext}\}\right)} \tag{4.24}$$

Using the definition of $\{\delta \boldsymbol{D}\}^I = [\boldsymbol{K}_t]^{-1}\{\hat{\boldsymbol{F}}_{ext}\}$ from Equation 4.13, it is possible to determine the predictor step for $\Delta\lambda$ as in Equation 4.25.

$$\Delta\lambda_1^n = \pm\frac{\Delta l}{\sqrt{\left((\{\delta \boldsymbol{D}\}_i^I)^T(\{\delta \boldsymbol{D}\}_i^I) + \beta^2\{\hat{\boldsymbol{F}}_{ext}\}^T\{\boldsymbol{F}_{ext}\}\right)}} \tag{4.25}$$

From Equation 4.25 it is evident, that there are two options when determining the predictor step for the load-level parameter. According to Crisfield (2000), the plus sign is to be used if the tangent stiffness $[\boldsymbol{K}_t]$ is positive definite and the negative sign is to be used if one of the eigenvalues are negative. A negative eigenvalue implies that a limit point has been passed.

However, to implement the geometric arc-length solver in the MATLAB Framework, another approach must be taken. The stiffness matrix in the MATLAB framework is a large sparse matrix, and determining eigenvalues is a demanding numerical problem. Additionally, the stiffness matrix in the MATLAB framework is non-symmetric, which further complicates the problem of determining eigenvalues.

Instead a method outlined in Geers (1999) is used. In this method the sign is determined according to Equation 4.26.

$$\Delta\lambda_1^n = \begin{cases} +|\Delta\lambda_1^n| & \text{if } (\{\Delta \boldsymbol{D}\}^{n-1})^T\{\delta \boldsymbol{D}^I\}_0^n > 0 \\ -|\Delta\lambda_1^n| & \text{if } (\{\Delta \boldsymbol{D}\}^{n-1})^T\{\delta \boldsymbol{D}^I\}_0^n < 0 \end{cases} \tag{4.26}$$

where $\{\Delta \boldsymbol{D}\}^{n-1}$ is the total displacement increment in the previous converged substep and $\{\delta \boldsymbol{D}^I\}_0^n$ is calculated according to Equation 4.13.

4.2 Matlab Implementation and Validation of Geometric Arc-Length Method

Based on the theory explained in the previous section, a Crisfield geometric arc-length solver is programmed and implemented in MATLAB. The algorithm is shown in Algorithm

1. The programmed arc-length solver is validated against three benchmarks examples in a step-wise manner in order to gradually build up the complexity of the solver. This is done in parallel with the development of the arc-length solver, while verifying it along the way. First a 1 DOF benchmarks example is considered and afterwards two 2 DOF's benchmarks examples are used to validate the solver. The validation against the benchmark examples is shown in Appendix C. Finally the arc-length solver is implemented in the MATLAB Framework and validated against a displacement controlled solution. In order to do so the algorithm shown in Algorithm 1 is modified to work in the MATLAB Framework, see Appendix D.

Algorithm 1 Simple Crisfield arc-length solver

Define solver settings, external loads and residual tolerance,
Initialize substep counter, displacement vector $\{\boldsymbol{D}\}$ and load parameter vector λ
while $\lambda\{\hat{\boldsymbol{F}}_{ext}\} < \{\boldsymbol{F}_{ext}\}$ **do**
  Update substep counter
  Evaluate tangent stiffness $[\boldsymbol{K}_t]$
  Calculate displacement correction $\{\delta\boldsymbol{D}^I\}_1^n$ according to Equation 4.13
  Calculate $\Delta\lambda_1$ according to Equation 4.25
  Calculate displacement increment $\{\Delta\boldsymbol{D}\}_1^n$ according to Equation 4.22
  Update displacement according to $\{\boldsymbol{D}\}_1^n = \{\boldsymbol{D}\}^{n-1} + \{\Delta\boldsymbol{D}\}_1^n$
  Update load-level parameter according to $\lambda_1^n = \lambda^{n-1} + \Delta\lambda_1^n$
  Calculate internal force $\{\boldsymbol{F}_{int}\}$
  Calculate the residual force $\{\boldsymbol{R}\}$ according to Equation 4.2
  Initiate iteration counter $icount = 0$
  while Residual tolerance $<$ residual force **do**
    Update iteration counter: $icount = icount + 1$
    Evaluate tangent stiffness $[\boldsymbol{K}_t]$
    Calculate displacement corrections $\{\delta\boldsymbol{D}^I\}_i^n$ and $\{\delta\boldsymbol{D}^{II}\}_i^n$ according to Equation 4.13
    Solve equation Equation 4.19 for $\delta\lambda_i^n$ and choose the correct root
    Calculate displacement correction $\{\delta\boldsymbol{D}\}_i^n$ according to Equation 4.12
    Update incremental displacement, $\{\Delta\boldsymbol{D}\}$ according to Equation 4.14
    Update incremental load level parameter $\Delta\lambda$ according to Equation 4.15
    Update displacement according to $\{\boldsymbol{D}\}_{i+1}^n = \{\boldsymbol{D}\}^{n-1} + \{\Delta\boldsymbol{D}\}_{i+1}^n$
    Update load-level parameter according to $\lambda_{i+1}^n = \lambda^{n-1} + \Delta\lambda_{i+1}^n$
    Calculate internal forces $\{\boldsymbol{F}_{int}\}$
    Calculate the residual force $\{\boldsymbol{R}\}$ according to Equation 4.2
  end while
end while

The implementation of the Crisfield arc-length solver in the MATLAB Framework is validated by solving a model, with the original displacement controlled Newton-Raphson solver. This solution is compared with one solved with the Crisfield arc-length solver. The model used for the validation is a DCB-test specimen with dimensions and boundary conditions as illustrated in Figure 4.3. In the displacement controlled solution a displacement is applied, whereas in the arc-length solver a force is applied. In both cases the analysis is stopped, when a displacement of 20 mm is reached in the node where the displacement/force is applied. This problem is chosen as it features both single and multiple delaminations, which means both snap-through and snap-back behaviour is expected to

occur in the model.

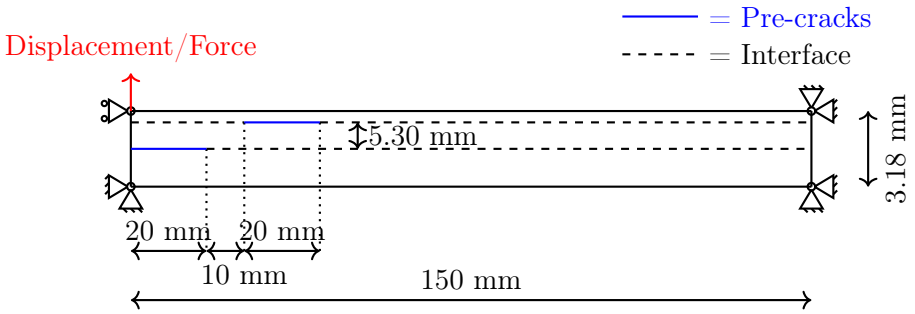


Figure 4.3. *Model used to validate the implementation of the arc-length solver in the Matlab framework*

Model and ARS-specific variables from the analysis are evident from Table 4.1. The material data used are evident from Table 4.2.

Element variables	Number
A-FNM elements (longitudinal)	90
A-FNM elements (thickness)	1
R_{Zone1}	1
R_{Zone23}	20
R_{Zone4}	1
N_{Zone3}	2
N_{Zone4}	90

Table 4.1. *Element and ARS-specific variables used in the model for validation of the Crisfield arc-length solver.*

Material parameter	Value
E_{11}	115.0 GPa
E_{22}	8.5 GPa
ν_{12}	0.0257
G_{12}	4.5 GPa
G_{Ic}	330 J/m^2
G_{IIc}	800 J/m^2
τ_{I0}	16.5 MPa
τ_{II0}	35.0 MPa
η	2.0
K_0	30 000 GPa

Table 4.2. *Material data used in the model for validation of the Crisfield arc-length solver.*

In Figure 4.4 the results from each test done on the Crisfield arc-length solver is illustrated and each substeb is highlighted with a marker according to the legend in the figure. The arc-length is changed from a small step length, yielding a fine discretisation of the load-displacement curve, to a rather large step length, yielding a coarse discretisation of the curve. Additionally, a test has been done with a step size of 0.0001 and a residual of both 1×10^{-4} N and 1×10^{-8} N. However, these are not plotted in the figure, as the small distance between each substep makes it difficult to see the results for the remaining tests.

Furthermore, the residual tolerance is investigated. Specifically a residual of 1×10^{-4} N and 1×10^{-8} N are tested. A residual of 1×10^{-4} N is used as it yields short analysis times and 1×10^{-8} N is checked to verify a solution can be obtained for even smaller residual errors.

The reason for checking these parameters in the arc-length solver is to validate the robustness of the final implementation. This is motivated by the fact that bugs in the code were found to cause convergence problems, either by utilizing extra iterations or diverging. However, based on the results in Figure 4.4 it is concluded that the arc-length

solver works satisfactory.

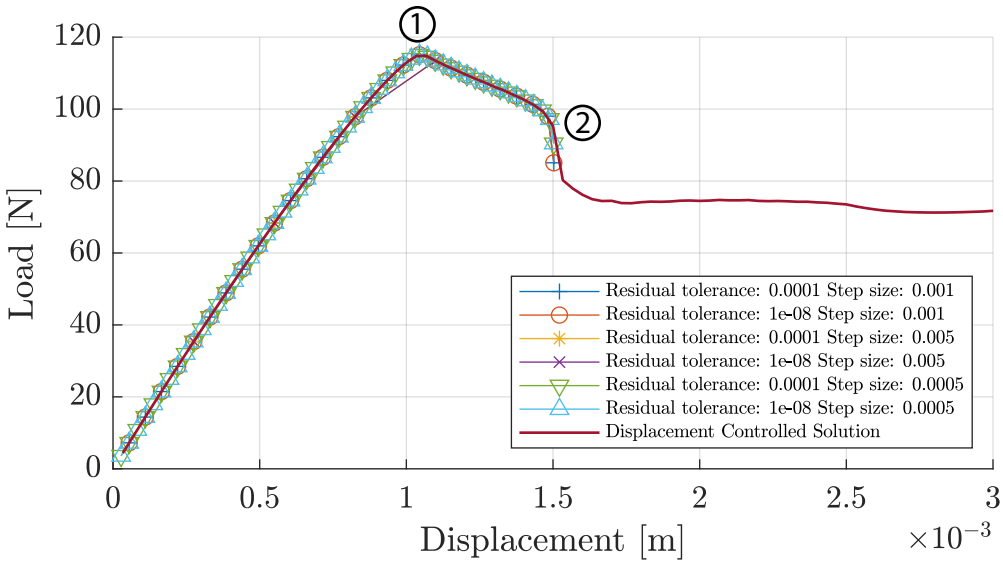


Figure 4.4. *Validation of Crisfield arc-length solver. All test are run with* $\beta = 0$.

The step size in each test in Figure 4.4 is kept constant throughout the entire analysis. The figure shows that all the tests traces the displacement controlled solution satisfactory. However, if the step size becomes too large, i.e. 0.005 , the peak of the load displacement curve is missed. Additionally, a step size of 0.005 causes the Crisfield arc-length solver to stop just after point (1) is passed, whereas the remaining tests successfully traces the curve until point (2).

During the implementation of the arc-length solver it is found that the ARS-related variables defined in subsection 3.2.1 greatly affects the convergence behaviour of the arc-length solver. Throughout the implementation of the arc-length solver this has been an additional unknown, which has made debugging troublesome. If too few sub-elements are used in the high gradient region, the Crisfield solver stopped prematurely, near point (1) in Figure 4.4. In this case the solver iteratively finds the same 2-4 points on the arc-length constraint, without advancing further. It has also been tried to increase the number of sub-elements in the refined zone, in order to make the solver advance past point (2), but this did not work.

In point (2) the second pre-crack in the model opens up and causes the Crisfield arc-length solver to diverge or outputting imaginary roots when solving Equation 4.19. In order to circumvent this problem a condition which automatically bisects the arc-length if the residual error becomes too large, if no real roots are found or if the number of iterations within the substep exceeds a predefined value, is implemented. However, this did not solve the problem as the arc-length solver kept bisecting the arc-length infinitely. According to de Borst et al. (2012) this is to be expected, as the Crisfield arc-length solver has problems when local material instability, in the form of the second pre-crack, occurs in the model. This can be solved by using other path-following constraint as explained in the next section.

Energy Dissipation Based Arc-Length Solvers

5

As mentioned, the Crisfield arc-length method, introduced in the previous section, uses a global geometrical path-following constraint. The geometrical constraint takes all degrees of freedom into account when the next point on the load-displacement curve is to be found. However, such a constraint is not applicable for models with a high degree of local material instability, such as delamination, and may fail to converge (de Borst et al., 2012). This inadequacy of not being able to model local instability, is believed to be the reason why the Chrisfield arc-length method is not able to solve for equilibrium points past point (2) in Figure 4.4.

When local material instability occurs, the degrees of freedom located in the vicinity of the local material instability becomes dominant for the continued deformation of the structure. For this reason the global constraint fails to accurately represent the instability (Börjesson et al., 2022).

Several approaches have been suggested in the literature to circumvent this problem. One approach is to only consider the degrees of freedom near the material instability, when defining the path following constraint. However, this implies the location of the material instability is known in advance and throughout the duration of the solution, reducing its generality. (Verhoosel et al., 2009)

Alternatively, Gutierrez (2004) formulated a path following constraint, based on the dissipated energy. The dissipated energy is related to the material instability in the model, making it ideal for a path-following constraint. When defining a path-following constraint, the constraint must monotonically increase in order for the solution to advance forward and fulfill the constraint equation $g = 0$. The rate of dissipated energy is always non-negative according to the second law of thermodynamics making the dissipated energy constant or increasing. (Verhoosel et al., 2009).

However, if there is no damage evolution in the model, the rate of dissipation is zero, which means this path-following constraint cannot be used. In such cases it is necessary to change the path-following constraint to e.g. the constraint defined in Equation 4.19.

The path-following constraint formulated by Gutierrez (2004) is done in terms of the global displacement and force vector for geometrically linear models with damage. Verhoosel et al. (2009) later extended Gutierrez (2004) method by deriving path-following constraints for other problems. In order to do so, new global vector quantities were introduced in addition to the terms used by Gutierrez (2004).

The approach taken by Gutierrez (2004) and Verhoosel et al. (2009) has its limitations, as the path-following constraint must be defined and changed according to the specific

problem. This reduces its generality. Secondly, some dissipation mechanism are difficult or impossible to define, in terms of the global displacement and force vectors, which further compromises the generality of this method.

With offset in the methods proposed by Gutierrez (2004) and Verhoosel et al. (2009), Börjesson et al. (2022) proposed another dissipation based arc-length solver, where the dissipated energy is calculated locally in each element and summed up over the entire domain. This is contrary to Gutierrez (2004) and Verhoosel et al. (2009) who does it at a global level.

Calculating the dissipated energy in each element makes it possible to generalise the path-following constraint. Multiple dissipation mechanisms can be combined in this method by adding the contribution from each mechanism. Additionally, defining the path-following constraint at a local level allows for including dissipation mechanism, which cannot be expressed in terms of global quantities. (Börjesson et al., 2022)

In this work two energy dissipation based arc-length solvers are implemented in the MATLAB Framework:

1. An arc-length solver based on Börjesson et al. (2022), referred to as the *local energy arc-length solver*
2. An arc-length solver based on Gutierrez (2004) and Verhoosel et al. (2009), referred to as the *global energy arc-length solver*

5.1 Local Energy Arc-Length Solver

This section takes offset in Börjesson et al. (2022) and explains the theory of the local energy arc-length solver. This is followed by a derivation of analytical terms, which are needed to implement the solver, in the MATLAB Framework. Finally, the arc-length solver is validated against a displacement controlled solution.

The total rate of dissipation of a body, G, is found by equation Equation 5.1, where $\dot{Q}$ is the specific rate of dissipation, and Ω is the area of the body.

$$G = \int_{\Omega} \dot{Q} \, d\Omega \tag{5.1}$$

The path-following constraint, g, is found by time-discretising Equation 5.1 with a backward Euler scheme, and introducing a parameter, $\Delta\tau$, which defines the amount of dissipated energy in the current substep of the solution. This is done in Equation 5.2.

$$g = \int_{\Omega} \Delta Q \, d\Omega - \Delta\tau = 0 \tag{5.2}$$

ΔQ describes the energy dissipation between two substeps. The definition of the path-following constraint in Equation 5.2, means $\Delta\tau$ can be interpreted as an equivalent to the step size, Δl, in the geometric arc-length solver in section 4.1.

The exact definition of $\dot{Q}$, and thereby also ΔQ, depends on the specific dissipating mechanism and the material model used. In this work, the dissipation mechanism in the models is related to opening and propagation of delamination cracks. This is modelled with cohesive elements in the MATLAB Framework as explained in chapter 3. This implies

that energy is only dissipated in the cohesive elements. The specific rate of dissipation, $\dot{Q}$, in the cohesive elements used in the MATLAB Framework is expressed by Equation 5.3 (Börjesson et al., 2022).

$$\dot{Q} = \frac{1}{2}\dot{d}\{\boldsymbol{s}_L\}^T[\boldsymbol{E}_p]\{\boldsymbol{s}_L\} \tag{5.3}$$

Where $\{\boldsymbol{s}_L\}$ is the separation vector defined in local coordinates in the cohesive interface, $[\boldsymbol{E}_p]$ is the penalty stiffness of the interface and d is the damage parameter that is used in the formulation of the cohesive element. For further details on the definitions of the cohesive element related variables, see Appendix A.

Time-discretising Equation 5.3 yields Equation 5.4.

$$\Delta Q = \frac{1}{2}\Delta d\{\boldsymbol{s}_L\}^T[\boldsymbol{E}_p]\{\boldsymbol{s}_L\} \tag{5.4}$$

By inserting Equation 5.4 in Equation 5.2 the path-following constraint that is used in the local energy arc-length solver is defined by Equation 5.5.

$$g = \int_{\Gamma_{cz}} \frac{1}{2}\Delta d\{\boldsymbol{s}_L\}^T[\boldsymbol{E}_p]\{\boldsymbol{s}_L\}\ d\Gamma - \Delta\tau = 0 \tag{5.5}$$

In the above equation the integral is changed to cover only the cohesive domain, Γ_{cz}, as energy is only dissipated in this domain. Due to the discretion of a finite element model, the integral in Equation 5.5 are evaluated individually over the area of each element, and summed up to yield the total dissipated energy, as in Equation 5.6.

$$\int_{\Gamma_{cz}} \frac{1}{2}\Delta d\{\boldsymbol{s}_L\}^T[\boldsymbol{E}_p]\{\boldsymbol{s}_L\}\ d\Gamma = \sum_{k=1}^{n} \int_{\Gamma_{cz_k}} \frac{1}{2}\Delta d\{\boldsymbol{s}_L\}^T[\boldsymbol{E}_p]\{\boldsymbol{s}_L\}\ d\Gamma \tag{5.6}$$

Where Γ_{cz_k} is the area of the k'th cohesive zone element and n is the total number of cohesive elements.

The arc-length solver is implemented with a method based on the same set of equations as used in the Riks arc-length method i.e. Equation 4.8 and 4.9 which are repeated below. Further details on Riks arc-length method is explained in Appendix B.

$$\begin{bmatrix} [\boldsymbol{K}_t]_i^n & -\{\hat{\boldsymbol{F}}_{ext}\} \\ (\{\boldsymbol{h}\}_i^n)^T & w_i^n \end{bmatrix} \begin{Bmatrix} \{\delta \boldsymbol{D}\}_i^n \\ \delta\lambda_i^n \end{Bmatrix} = \begin{Bmatrix} \{\boldsymbol{R}\}_i^n \\ -g_i^n \end{Bmatrix} \tag{4.8}$$

$$[\boldsymbol{K}_t]_i^n = \frac{\partial\{\boldsymbol{F}_{int}\}}{\partial\{\boldsymbol{D}\}} \qquad \{\boldsymbol{h}\}_i^n = \frac{\partial g}{\partial\{\boldsymbol{D}\}} \qquad w_i^n = \frac{\partial g}{\partial\lambda} \tag{4.9}$$

Thereby, expressions for the constraint derivatives, $\{\boldsymbol{h}\}_i^n$ and w_i^n are needed. Due to the definition of the path-following constraint, g, in Equation 5.5, g is independent of the load-level parameter, λ, making w equal to 0. Each entry in, $\{\boldsymbol{h}\}_i^n$, is expressed by Equation 5.7.

$$h_j = \int_{\Gamma_{cz_k}} \left\{\frac{\partial\Delta Q}{\partial\{\boldsymbol{s}_L\}}\right\} \left\{\frac{\partial\{\boldsymbol{s}_L\}}{\partial d_j}\right\}\ d\Gamma \tag{5.7}$$

Where Γ_{cz_k} is the cohesive domain of the k'th cohesive element. The global vector, $\{\boldsymbol{h}\}$, is assembled similarly to the internal force vector in the MATLAB Framework and this is done for each iteration.

The integral in Equation 5.6 and Equation 5.7 are both integrated by numerical integration, as this is easily implemented in the MATLAB Framework. The MATLAB Framework has both Newton-Cotes integration and Gauss integration available. In this work, a two point Gauss quadrature is used. As mentioned in chapter 3, the cohesive element used in the MATLAB Framework is a 2D cohesive element. Therefore, the element area is integrated over the length, L, and multiplied by its thickness, t as done in Equation 5.8 and 5.9.

$$\int_{\Gamma_{cz_k}} \frac{1}{2}\Delta d\{\boldsymbol{s}_L\}^T[\boldsymbol{E}_p]\{\boldsymbol{s}_L\}\ d\Gamma = \int_{x=0}^{x=L} \frac{1}{2}\Delta d\{\boldsymbol{s}_L\}^T[\boldsymbol{E}_p]\{\boldsymbol{s}_L\}t\ dx \tag{5.8}$$

$$\int_{\Gamma_{cz_k}} \left\{\frac{\partial \Delta Q}{\partial\{\boldsymbol{s}_L\}}\right\}\left\{\frac{\partial\{\boldsymbol{s}_L\}}{\partial d_j}\right\}\ d\Gamma = \int_{x=0}^{x=L} \left\{\frac{\partial \Delta Q}{\partial\{\boldsymbol{s}_L\}}\right\}\left\{\frac{\partial\{\boldsymbol{s}_L\}}{\partial d_j}\right\} t\ dx \tag{5.9}$$

In order to use a two point Gauss quadrature the integral limits must be changed from [0,L] to [-1,1]. This is done by changing from global coordinates to local coordinates in the element by using the Jacobian determinant, J.

$$\int_{x=0}^{x=L} \frac{1}{2}\Delta d\{\boldsymbol{s}_L\}^T[\boldsymbol{E}_p]\{\boldsymbol{s}_L\}t\ dx = \int_{\xi=-1}^{\xi=1} \frac{1}{2}\Delta d\{\boldsymbol{s}_L\}^T[\boldsymbol{E}_p]\{\boldsymbol{s}_L\}tJ\ d\xi \tag{5.10}$$

$$\int_{x=0}^{x=L} \left\{\frac{\partial \Delta Q}{\partial\{\boldsymbol{s}_L\}}\right\}\left\{\frac{\partial\{\boldsymbol{s}_L\}}{\partial d_j}\right\} t\ dx = \int_{\xi=-1}^{\xi=1} \left\{\frac{\partial \Delta Q}{\partial\{\boldsymbol{s}_L\}}\right\}\left\{\frac{\partial\{\boldsymbol{s}_L\}}{\partial d_j}\right\} tJ\ d\xi \tag{5.11}$$

In general for a function, $\phi(\xi)$, Gauss integration is performed by evaluating the function in specific predefined points and multiplying with predefined weight factors. For a two point Gauss quadrature this is done according to Equation 5.12:

$$I = \int_{-1}^{1} \phi\ d\xi \approx W_1\phi_1 + W_2\phi_2 \tag{5.12}$$

where W_1 and W_2 are weight factors, both equal to 1, and ϕ_1 and ϕ_2 are the function value evaluated in the points $\xi = \frac{-1}{\sqrt{3}}$ and $\xi = \frac{1}{\sqrt{3}}$, respectively.

5.1.1 Derivation of Analytical Terms for The Arc-Length Solver

In order to implement the local energy dissipation based arc-length solver in the MATLAB Framework, expression for each of the terms $\left\{\frac{\partial \Delta Q}{\partial\{\boldsymbol{s}_L\}}\right\}$ and $\left\{\frac{\partial\{\boldsymbol{s}_L\}}{\partial d_j}\right\}$ in Equation 5.7 needs to be derived. Expression for the first term is defined in Equation 5.13.

$$\left\{\frac{\partial \Delta Q}{\partial\{\boldsymbol{s}_L\}}\right\} = \Delta d\{\boldsymbol{s}_L\}^T[\boldsymbol{E}_p] \tag{5.13}$$

An expression for the second term is more comprehensive, as it requires a derivation of the separation vector in the cohesive element. A derivation of the separation vector is given in the following section. This derivation takes offset in Camanho et al. (2003) and Turon et al. (2006).

The cohesive element used in the MATLAB Framework is a 4-node zero thickness element and each node has two degrees of freedom, one in the x-direction and one in the y-direction. The displacement field of the upper, $\{\tilde{\boldsymbol{u}}^+\}$, and lower, $\{\tilde{\boldsymbol{u}}^-\}$, crack faces are expressed by

the nodal displacements, $\{\boldsymbol{d}\}$ and shape functions, $[\boldsymbol{N}]$ as done in Equation 5.14 and Equation 5.15.

$$\{\tilde{\boldsymbol{u}}^+\} = \begin{bmatrix}\tilde{u}^+(x,y)\\ \tilde{v}^+(x,y)\end{bmatrix} = [\boldsymbol{N}]\{\boldsymbol{d}^+\} = \begin{bmatrix}N_1 & 0 & N_2 & 0\\ 0 & N_1 & 0 & N_2\end{bmatrix}\begin{bmatrix}u_1^+\\ v_1^+\\ u_2^+\\ v_2^+\end{bmatrix} \tag{5.14}$$

$$\{\tilde{\boldsymbol{u}}^-\} = \begin{bmatrix}\tilde{u}^-(x,y)\\ \tilde{v}^-(x,y)\end{bmatrix} = [\boldsymbol{N}]\{\boldsymbol{d}^-\} = \begin{bmatrix}N_1 & 0 & N_2 & 0\\ 0 & N_1 & 0 & N_2\end{bmatrix}\begin{bmatrix}u_1^-\\ v_1^-\\ u_2^-\\ v_2^-\end{bmatrix} \tag{5.15}$$

$\{\boldsymbol{d}^+\}$ and $\{\boldsymbol{d}^-\}$ contains nodal displacements for the top and bottom crack face, respectively.

The shape functions are expressed in Equation 5.16, in terms of the local coordinates ξ.

$$N_1 = \frac{(1-\xi)}{2} \qquad N_2 = \frac{(1+\xi)}{2} \tag{5.16}$$

Based on the above expression for the displacement fields of the crack faces in the cohesive element, it is possible to express the separation vector in global coordinates, $\{\boldsymbol{s}_G\}$, as the difference in the displacements fields. This is done in Equation 5.17.

$$\{\boldsymbol{s}_G\} = \{\tilde{\boldsymbol{u}}^+\} - \{\tilde{\boldsymbol{u}}^-\} = [\boldsymbol{N}]\{\boldsymbol{d}^+\} - [\boldsymbol{N}]\{\boldsymbol{d}^-\} \tag{5.17}$$

It is possible to redefine the separation vector in global coordinates by defining the shape functions according to the location of the nodes, as in Equation 5.18.

$$\bar{N}_p = \begin{cases} N_p & p \in \text{nodes located at the upper crack face}\\ -N_p & p \in \text{nodes located at the upper crack face}\end{cases} \tag{5.18}$$

Thereby, the displacement jump vector in global coordinates is found according to Equation 5.19.

$$\{\boldsymbol{s}_G\} = \{\tilde{\boldsymbol{u}}^+\} - \{\tilde{\boldsymbol{u}}^-\} = [\bar{\boldsymbol{N}}]\{\boldsymbol{d}\} = \begin{bmatrix}N_1 & 0 & -N_1 & 0 & N_2 & 0 & -N_2 & 0\\ 0 & N_1 & 0 & -N_1 & 0 & N_2 & 0 & -N_2\end{bmatrix}\begin{bmatrix}u_1^+\\ v_1^+\\ u_1^-\\ v_1^-\\ u_2^+\\ v_2^+\\ u_2^-\\ v_2^-\end{bmatrix} \tag{5.19}$$

The separation vector in local coordinates, $\{\boldsymbol{s}_L\}$, is expressed as the separation vector in the global coordinate system, $\{\boldsymbol{s}_G\}$, multiplied by a rotation matrix, $[\boldsymbol{T}]$. Thereby, the separation vector in local coordinates are determined by Equation 5.20.

$$\{\boldsymbol{s}_L\} = [\boldsymbol{T}]\{\boldsymbol{s}_G\} = [\boldsymbol{T}][\bar{\boldsymbol{N}}]\{\boldsymbol{d}\} = [\boldsymbol{T}]\begin{bmatrix}N_1u_1^+ - N_1u_1^- + N_2u_2^+ - N_2u_2^-\\ N_1v_1^+ - N_1v_1^- + N_2v_2^+ - N_2v_2^-\end{bmatrix} \tag{5.20}$$

In order to fully describe the separation vector by analytical expressions, an expression for each term in the rotation matrix needs to be derived. The rotation matrix is determined by the direction cosines, between a tangent vector and a normal vector to the midsurface of the crack face. The tangent vector to the crack face, $\{\nu\}$, is determined by differentiating the global position vector of the crack midsurface, $\{\bar{\boldsymbol{x}}\}$, with respect to the local element coordinate, ξ.

Also, the geometry, $\{\boldsymbol{x}\}$, of an finite element is generally defined by shape functions, $[\boldsymbol{N}]$, and nodal coordinates, $\{\boldsymbol{c}\}$, as described by Equation 5.21.

$$\{\boldsymbol{x}\} = [\boldsymbol{N}]\{\boldsymbol{c}\} \tag{5.21}$$

The nodal coordinates of the midsurface are defined by Equation 5.22, where x_p and y_p, are the undeformed x- and y-coordinate of the crack faces and, u and v are nodal displacements in the x- and y-direction, respectively.

$$\{\boldsymbol{c}\} = \left(\begin{bmatrix} x_{p1}^{+} \\ y_{p1}^{+} \\ x_{p2}^{+} \\ y_{p2}^{+} \end{bmatrix} + \begin{bmatrix} x_{p1}^{-} \\ y_{p1}^{-} \\ x_{p2}^{-} \\ y_{p2}^{-} \end{bmatrix} + \begin{bmatrix} u_{1}^{+} \\ v_{1}^{+} \\ u_{2}^{+} \\ v_{2}^{+} \end{bmatrix} + \begin{bmatrix} u_{1}^{-} \\ v_{1}^{-} \\ u_{2}^{-} \\ v_{2}^{-} \end{bmatrix} \right) \tag{5.22}$$

Thereby, the midsurface of the crack is interpolated by Equation 5.23.

$$\{\bar{\boldsymbol{x}}\} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{2} \begin{bmatrix} N_1 & 0 & N_2 & 0 \\ 0 & N_1 & 0 & N_2 \end{bmatrix} \left(\begin{bmatrix} x_{p1}^{+} \\ y_{p1}^{+} \\ x_{p2}^{+} \\ y_{p2}^{+} \end{bmatrix} + \begin{bmatrix} x_{p1}^{-} \\ y_{p1}^{-} \\ x_{p2}^{-} \\ y_{p2}^{-} \end{bmatrix} + \begin{bmatrix} u_{1}^{+} \\ v_{1}^{+} \\ u_{2}^{+} \\ v_{2}^{+} \end{bmatrix} + \begin{bmatrix} u_{1}^{-} \\ v_{1}^{-} \\ u_{2}^{-} \\ v_{2}^{-} \end{bmatrix} \right) \tag{5.23}$$

The shape functions are as defined in Equation 5.16.

As mentioned above, the tangent vector to the midsurface of the crack, $\{\nu\}$, is expressed by differentiation of the global position vector with respect to the local coordinate, ξ. Further, only the shape functions are functions of the local coordinate, ξ, which simplifies differentiation of Equation 5.23 to Equation 5.24.

$$\{\boldsymbol{\nu}\} = \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} = \frac{1}{2} \frac{\partial[\boldsymbol{N}]}{\partial \xi} \left(\begin{bmatrix} x_{p1}^{+} \\ y_{p1}^{+} \\ x_{p2}^{+} \\ y_{p2}^{+} \end{bmatrix} + \begin{bmatrix} x_{p1}^{-} \\ y_{p1}^{-} \\ x_{p2}^{-} \\ y_{p2}^{-} \end{bmatrix} + \begin{bmatrix} u_{1}^{+} \\ v_{1}^{+} \\ u_{2}^{+} \\ v_{2}^{+} \end{bmatrix} + \begin{bmatrix} u_{1}^{-} \\ v_{1}^{-} \\ u_{2}^{-} \\ v_{2}^{-} \end{bmatrix} \right) \tag{5.24}$$

The partial derivative of the shape functions are defined in Equation 5.25.

$$\frac{\partial[\boldsymbol{N}]}{\partial \xi} = \begin{bmatrix} \frac{-1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{-1}{2} & 0 & \frac{1}{2} \end{bmatrix} \tag{5.25}$$

Combining equation Equation 5.25 and Equation 5.24 yields an expression for the tangent vector to the midsurface of the crack. This is done in Equation 5.26.

$$\begin{aligned} \{\boldsymbol{\nu}\} = \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} \frac{-1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{-1}{2} & 0 & \frac{1}{2} \end{bmatrix} \left(\begin{bmatrix} x_{p1}^{+} + x_{p1}^{-} + u_{1}^{+} + u_{1}^{-} \\ y_{p1}^{+} + y_{p1}^{-} + v_{1}^{+} + v_{1}^{-} \\ x_{p2}^{+} + x_{p2}^{-} + u_{2}^{+} + u_{2}^{-} \\ y_{p2}^{+} + y_{p2}^{-} + v_{2}^{+} + v_{2}^{-} \end{bmatrix} \right) \\ &= \frac{1}{4} \begin{bmatrix} -1(x_{p1}^{+} + x_{p1}^{-} + u_{1}^{+} + u_{1}^{-}) + (x_{p2}^{+} + x_{p2}^{-} + u_{2}^{+} + u_{2}^{-}) \\ -1(y_{p1}^{+} + y_{p1}^{-} + v_{1}^{+} + v_{1}^{-}) + (y_{p2}^{+} + y_{p2}^{-} + v_{2}^{+} + v_{2}^{-}) \end{bmatrix} \end{aligned} \tag{5.26}$$

The tangent vector is normalised.

$$\{\boldsymbol{\nu}_{\text{norm}}\} = \frac{\{\boldsymbol{\nu}\}}{||\{\boldsymbol{\nu}\}||} = \begin{Bmatrix} \nu_{1,norm} \\ \nu_{2,norm} \end{Bmatrix} \tag{5.27}$$

where $\nu_{1,norm}$ and $\nu_{2,norm}$ are defined in Equation 5.28 and 5.29.

$$\nu_{1,\text{norm}} = \frac{-\frac{1}{4}(x_{p1}^{+}+x_{p1}^{-}+u_1^{+}+u_1^{-})+\frac{1}{4}(x_{p2}^{+}+x_{p2}^{-}+u_2^{+}+u_2^{-})}{\sqrt{\left[-\frac{1}{4}(x_{p1}^{+}+x_{p1}^{-}+u_1^{+}+u_1^{-})+\frac{1}{4}(x_{p2}^{+}+x_{p2}^{-}+u_2^{+}+u_2^{-})\right]^2+\left[-\frac{1}{4}(y_{p1}^{+}+y_{p1}^{-}+v_1^{+}+v_1^{-})+\frac{1}{4}(y_{p2}^{+}+y_{p2}^{-}+v_2^{+}+v_2^{-})\right]^2}} \tag{5.28}$$

$$\nu_{2,\text{norm}} = \frac{-\frac{1}{4}(y_{p1}^{+}+y_{p1}^{-}+v_1^{+}+v_1^{-})+\frac{1}{4}(y_{p2}^{+}+y_{p2}^{-}+v_2^{+}+v_2^{-})}{\sqrt{\left[-\frac{1}{4}(x_{p1}^{+}+x_{p1}^{-}+u_1^{+}+u_1^{-})+\frac{1}{4}(x_{p2}^{+}+x_{p2}^{-}+u_2^{+}+u_2^{-})\right]^2+\left[-\frac{1}{4}(y_{p1}^{+}+y_{p1}^{-}+v_1^{+}+v_1^{-})+\frac{1}{4}(y_{p2}^{+}+y_{p2}^{-}+v_2^{+}+v_2^{-})\right]^2}} \tag{5.29}$$

A normal vector to the crack surface, $\{\boldsymbol{\mu}\}$ can now be defined by interchanging the entrances in $\{\boldsymbol{\nu}\}$ and adding a minus sign for the first entrance. This is done in Equation 5.30.

$$\{\boldsymbol{\mu}\} = \{\hat{\boldsymbol{\nu}}_{\text{norm}}\} = \begin{bmatrix} -\nu_{2,\text{norm}} \\ \nu_{1,\text{norm}} \end{bmatrix} \tag{5.30}$$

The rotation matrix can now be defined from the tangent vector, $\{\boldsymbol{\nu}\}$, and the normal vector, $\{\boldsymbol{\mu}\}$, as in Equation 5.31.

$$[\boldsymbol{T}] = \begin{bmatrix} \nu_{1,\text{norm}} & \mu_1 \\ \nu_{2,\text{norm}} & \mu_2 \end{bmatrix} = \begin{bmatrix} \nu_{1,\text{norm}} & -\nu_{2,\text{norm}} \\ \nu_{2,\text{norm}} & \nu_{1,\text{norm}} \end{bmatrix} \tag{5.31}$$

Thereby, combining Equation 5.20 and 5.31 the full analytical expression for the separation vector, $\{\boldsymbol{s}_L\}$, in local coordinates is expressed by Equation 5.32.

$$\begin{aligned} \{\boldsymbol{s}_L\} &= \begin{bmatrix} \nu_{1,\text{norm}} & -\nu_{2,\text{norm}} \\ \nu_{2,\text{norm}} & \nu_{1,\text{norm}} \end{bmatrix} \begin{bmatrix} N_1u_1^{+} - N_1u_1^{-} + N_2u_2^{+} - N_2u_2^{-} \\ N_1v_1^{+} - N_1v_1^{-} + N_2v_2^{+} - N_2v_2^{-} \end{bmatrix} \\ &= \begin{bmatrix} \nu_{1,\text{norm}}[N_1u_1^{+} - N_1u_1^{-} + N_2u_2^{+} - N_2u_2^{-}] - \nu_{2,\text{norm}}[N_1v_1^{+} - N_1v_1^{-} + N_2v_2^{+} - N_2v_2^{-}] \\ \nu_{2,\text{norm}}[N_1u_1^{+} - N_1u_1^{-} + N_2u_2^{+} - N_2u_2^{-}] + \nu_{1,\text{norm}}[N_1u_1^{+} - N_1u_1^{-} + N_2u_2^{+} - N_2u_2^{-}] \end{bmatrix} \end{aligned} \tag{5.32}$$

It is now possible to derive expressions for $\left\{\frac{\partial\{\boldsymbol{s}_L\}}{\partial d_j}\right\}$, in Equation 5.7. This is not included in the report, due to the length of each expression, but can be found in Appendix E.

Thus, expression for all the terms related to the path-following constraint, needed to redefine Riks arc-length method as expressed in Equation 4.8 and 4.9 are now defined. Thereby, it is possible to implement the solver in the MATLAB Framework.

The solver algorithm implemented in the MATLAB Framework is available Appendix F. As mentioned in the introduction to this chapter, the arc-length solver presented in this section, requires that the energy dissipation between sub-steps is positive and for this reason the path-following constraint must be changed if there exist non-dissipative part on the load-displacement curve. This is e.g. the case in the initial linear elastic part of an analysis, which means the solution must be initialized with e.g. a geometric arc-length solver such as Crisfield or Riks arc-length solver.

5.1.2 Validation of Local Energy Arc-Length Solver

In this section an evaluation of the implementation of the local energy arc-length solver in the MATLAB Framework solver is done. The solver is successfully applied to solve

the same problem as in the validation of the Crisfield solver in section 4.2. The result is illustrated in Figure 5.1. The arc-length solver yields the same load-displacement curve as the displacement controlled solution. Furthermore, it captures the snap-back behaviour at point (2) in Figure 5.1.

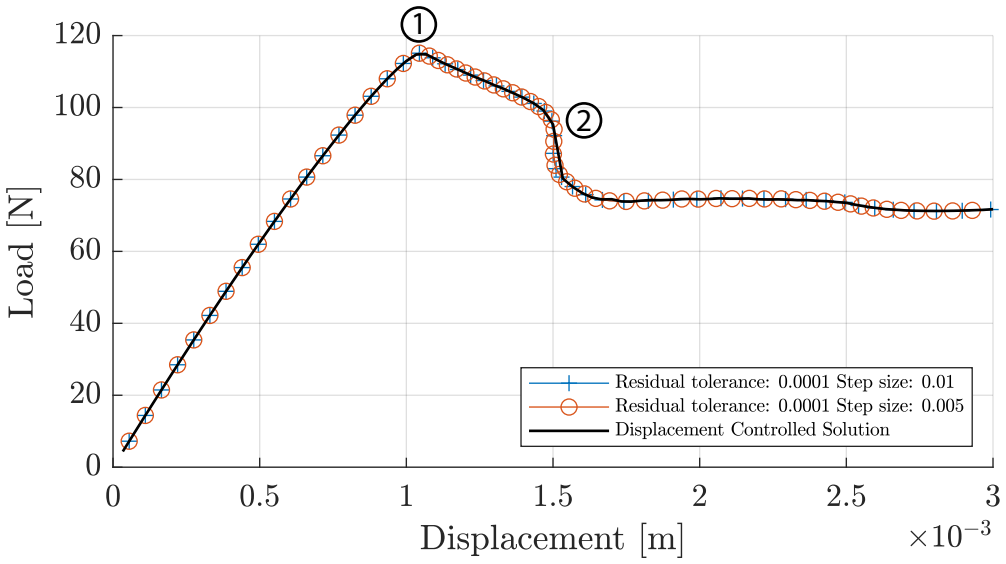


Figure 5.1. *Load-displacement curve used to validate the local energy arc-length solver against a displacement controlled solution. Crisfields arc length solver is used to initiate the solution and the shift to the local energy arc-length solver happens at point 1.*

However, tests of the solver has revealed problems related to the robustness. Even though a substantial amount of time has been put into debugging the implementation of the arc-length solver, the problem has not been fully fixed.

The main problem encountered with the local energy arc-length solver is convergence difficulties due to the step size used.

In order to avoid using too large step-sizes, a condition that bisects the step size is implemented in the solver. The bisecting is similar to the bisecting condition done for the Crisfield arc-length solver in section 4.2. However this did not solve the convergence difficulties since both too small and to large step sizes cause difficulties.

The solver runs successfully if the step size, $\Delta\tau$, is limited to be within a small interval. For the analysis presented in Figure 5.1, the limit of $\Delta\tau$, is approximately [0.0005-0.01]. In aforementioned interval, the solver uses 5-25 iterations within each substep and the residual generally decreases monotonically for each iteration.

Using a step size just below the acceptable interval, generally makes the solver run satisfactory for a number of substeps, until the solver starts diverging. This ultimately causes the solver to automatically bisect the step size, due to the aforementioned condition. Bisecting the step size yields a new steps size, even further from the acceptable interval. This causes the solver to diverge even faster and this process keeps repeating.

It makes sense that the solver has convergence problems for large step sizes. If the step becomes too large, the linearisation of the equilibrium equation and the path-following constraint, done according to Equation 4.7, yields a poor fit to the actual curves which causes convergence problems.

The convergence problems related to utilizing smaller step sizes cannot be explained. Intuitively, reducing the step size should improve the convergence rate, as the linearisation should yield a better approximation to the actual curve. Nevertheless, this was not experienced with the solver.

In order to debug what cause the convergence issues several approaches is taken:

- Check if constraint g is fulfilled
- Test of implementation of structure of equations
- Central difference check on $\left\{\frac{\partial \Delta Q}{\partial \{\boldsymbol{s}_L\}}\right\}$ and $\left\{\frac{\partial \{\boldsymbol{s}_L\}}{\partial d_j}\right\}$ in the constraint derivative, $\{\boldsymbol{h}\}$
- Central difference check on the constraint derivative $\{\boldsymbol{h}\}$

The approaches and outcome is explained in the following:

Check if constraint g is fulfilled

First of, the dissipated energy in each converged substep is checked and compared with the magnitude of the step size, $\Delta\tau$. This is done to check whether the arc-length constraint is fulfilled for each converged substep. For all the different analyses checked, this is the case.

Test of implementation of structure of equations

As mentioned earlier, the local energy arc-length solver is based on the same set of equations as Riks arc-length method, but uses a different path-following constraint. An arc-length solver based on Riks arc-length method is implemented in the MATLAB Framework. This is done to check whether the implementation of the set of equations is correct. Details of Riks arc-length method and its implementation is explained in Appendix B.

The implementation of Riks arc-length method is done successfully. Thus, it is concluded that the error does not lie within the implementation of the set of equations. The error is instead expected to be found in the calculation of either the path-following constraint, g, in Equation 5.5 or the constraint derivative, $\{\boldsymbol{h}\}$ in Equation 5.7, which is where the Riks and local energy solver differ.

Central difference check on $\left\{\frac{\partial \Delta Q}{\partial \{\boldsymbol{s}_L\}}\right\}$ and $\left\{\frac{\partial \{\boldsymbol{s}_L\}}{\partial d_j}\right\}$ in the constraint derivative, $\{\boldsymbol{h}\}$

The definition of the constraint derivative, $\{\boldsymbol{h}\}$, is considerably more demanding than the path-following constraint, g, both in terms of the theoretical derivation and its implementation in the MATLAB Framework. Therefore, focus has been put on the $\{\boldsymbol{h}\}$ term.

To validate the analytical expressions derived for the constraint derivative, $\{\boldsymbol{h}\}$, central differences have been used to numerically calculate the terms. Specifically, central difference is used to calculate both $\left\{\frac{\partial \Delta Q}{\partial \{\boldsymbol{s}_L\}}\right\}$ and $\left\{\frac{\partial \{\boldsymbol{s}_L\}}{\partial d_j}\right\}$, individually.

The central differences are used as a quick check on the values of each of the aforementioned terms. Due to this and limited time, there has not been put much effort into determining the correct size of the perturbation. Instead the perturbation has been decreased in steps, until almost no change between two perturbations sizes is achieved. A perturbation of $1.0 \times 10^{-7}\,\mathrm{m}$ is used to calculate $\left\{\frac{\partial \Delta Q}{\partial \{\boldsymbol{s}_L\}}\right\}$. Similarly, a perturbation of $1.0 \times 10^{-6}\,\mathrm{m}$

is used to calculate $\left\{\frac{\partial\{\boldsymbol{s}_L\}}{\partial d_j}\right\}$. Almost identical results from the analytical and central difference approach are achieved. An example of the analytical and central difference results for a randomly selected element in substep 15 and 17 are given in Table 5.1.

Substep 15

	IP	$\frac{\partial\Delta Q}{\partial\{\boldsymbol{s}_L\}}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial u_1^+}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial v_1^+}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial u_1^-}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial v_1^-}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial u_2^-}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial v_2^-}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial u_2^+}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial v_2^+}$
Analytical:	1	$\begin{bmatrix}0.0032\cdot10^6\\3.2286\cdot10^6\end{bmatrix}$	$\begin{bmatrix}0.7887\\0.0014\end{bmatrix}$	$\begin{bmatrix}-0.0015\\0.6977\end{bmatrix}$	$\begin{bmatrix}-0.7887\\-0.0018\end{bmatrix}$	$\begin{bmatrix}0.0017\\-0.8796\end{bmatrix}$	$\begin{bmatrix}-0.2113\\-0.0002\end{bmatrix}$	$\begin{bmatrix}0.0003\\-0.1204\end{bmatrix}$	$\begin{bmatrix}0.2113\\0.0006\end{bmatrix}$	$\begin{bmatrix}-0.0005\\0.3023\end{bmatrix}$
	2	$\begin{bmatrix}0.0034\cdot10^6\\3.3526\cdot10^6\end{bmatrix}$	$\begin{bmatrix}0.2113\\0.0002\end{bmatrix}$	$\begin{bmatrix}-0.0003\\0.1254\end{bmatrix}$	$\begin{bmatrix}-0.2113\\-0.0006\end{bmatrix}$	$\begin{bmatrix}0.0005\\-0.2973\end{bmatrix}$	$\begin{bmatrix}-0.7887\\-0.0014\end{bmatrix}$	$\begin{bmatrix}0.0015\\-0.7027\end{bmatrix}$	$\begin{bmatrix}0.7887\\0.0018\end{bmatrix}$	$\begin{bmatrix}-0.0017\\0.8746\end{bmatrix}$
Central difference:	1	$\begin{bmatrix}0.0032\cdot10^6\\3.2286\cdot10^6\end{bmatrix}$	$\begin{bmatrix}0.7887\\0.0016\end{bmatrix}$	$\begin{bmatrix}-0.0016\\0.7887\end{bmatrix}$	$\begin{bmatrix}-0.7887\\-0.0016\end{bmatrix}$	$\begin{bmatrix}0.0016\\-0.7887\end{bmatrix}$	$\begin{bmatrix}-0.2113\\-0.0004\end{bmatrix}$	$\begin{bmatrix}0.0004\\-0.2113\end{bmatrix}$	$\begin{bmatrix}0.2113\\0.0004\end{bmatrix}$	$\begin{bmatrix}-0.0004\\0.2113\end{bmatrix}$
	2	$\begin{bmatrix}0.0034\cdot10^6\\3.3526\cdot10^6\end{bmatrix}$	$\begin{bmatrix}0.2113\\0.0004\end{bmatrix}$	$\begin{bmatrix}-0.0004\\0.2113\end{bmatrix}$	$\begin{bmatrix}-0.2113\\-0.0004\end{bmatrix}$	$\begin{bmatrix}0.0004\\-0.2113\end{bmatrix}$	$\begin{bmatrix}-0.7887\\-0.0016\end{bmatrix}$	$\begin{bmatrix}0.0016\\-0.7887\end{bmatrix}$	$\begin{bmatrix}0.7887\\0.0016\end{bmatrix}$	$\begin{bmatrix}-0.0016\\0.7887\end{bmatrix}$

Substep 17

	IP	$\frac{\partial\Delta Q}{\partial\{\boldsymbol{s}_L\}}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial u_1^+}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial v_1^+}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial u_1^-}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial v_1^-}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial u_2^-}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial v_2^-}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial u_2^+}$	$\frac{\partial\{\boldsymbol{s}_L\}}{\partial v_2^+}$
Analytical:	1	$\begin{bmatrix}0.0055\cdot10^6\\5.1646\cdot10^6\end{bmatrix}$	$\begin{bmatrix}0.7887\\0.0017\end{bmatrix}$	$\begin{bmatrix}-0.0018\\0.7360\end{bmatrix}$	$\begin{bmatrix}-0.7887\\-0.0020\end{bmatrix}$	$\begin{bmatrix}0.0019\\-0.8413\end{bmatrix}$	$\begin{bmatrix}-0.2113\\-0.0004\end{bmatrix}$	$\begin{bmatrix}0.0004\\-0.1587\end{bmatrix}$	$\begin{bmatrix}0.2113\\0.0006\end{bmatrix}$	$\begin{bmatrix}-0.0005\\0.2640\end{bmatrix}$
	2	$\begin{bmatrix}0.0062\cdot10^6\\5.4212\cdot10^6\end{bmatrix}$	$\begin{bmatrix}0.2113\\0.0004\end{bmatrix}$	$\begin{bmatrix}-0.0004\\0.1625\end{bmatrix}$	$\begin{bmatrix}-0.2113\\-0.0006\end{bmatrix}$	$\begin{bmatrix}0.0005\\-0.2601\end{bmatrix}$	$\begin{bmatrix}-0.7887\\-0.0017\end{bmatrix}$	$\begin{bmatrix}0.0018\\-0.7399\end{bmatrix}$	$\begin{bmatrix}0.7887\\0.0020\end{bmatrix}$	$\begin{bmatrix}-0.0019\\0.8375\end{bmatrix}$
Central difference:	1	$\begin{bmatrix}0.0055\cdot10^6\\5.1646\cdot10^6\end{bmatrix}$	$\begin{bmatrix}0.7887\\0.0019\end{bmatrix}$	$\begin{bmatrix}-0.0019\\0.7887\end{bmatrix}$	$\begin{bmatrix}-0.7887\\-0.0019\end{bmatrix}$	$\begin{bmatrix}0.0019\\-0.7887\end{bmatrix}$	$\begin{bmatrix}-0.2113\\-0.0005\end{bmatrix}$	$\begin{bmatrix}0.0005\\-0.2113\end{bmatrix}$	$\begin{bmatrix}0.2113\\0.0005\end{bmatrix}$	$\begin{bmatrix}-0.0005\\0.2113\end{bmatrix}$
	2	$\begin{bmatrix}0.0062\cdot10^6\\5.4212\cdot10^6\end{bmatrix}$	$\begin{bmatrix}0.2113\\0.0005\end{bmatrix}$	$\begin{bmatrix}-0.0005\\0.2113\end{bmatrix}$	$\begin{bmatrix}-0.2113\\-0.0005\end{bmatrix}$	$\begin{bmatrix}0.0005\\-0.2113\end{bmatrix}$	$\begin{bmatrix}-0.7887\\-0.0019\end{bmatrix}$	$\begin{bmatrix}0.0019\\-0.7887\end{bmatrix}$	$\begin{bmatrix}0.7887\\0.0019\end{bmatrix}$	$\begin{bmatrix}-0.0019\\0.7887\end{bmatrix}$

Table 5.1. *Comparison of analytical and central difference results for each term in the constraint derivative $\{\boldsymbol{h}\}$ in Equation 5.7. IP = Integration point.*

It is clear from Table 5.1, that the agreement between the analytical and central difference method is good. Even though the perturbation size has not been investigated thoroughly, it is assessed that the analytical derivation and implementation is correct due to the similarity in the results.

Central difference check on the constraint derivative $\{\boldsymbol{h}\}$

The next step in the process of locating the error, is to use central differences to calculate the constraint derivative, $\{\boldsymbol{h}\}$, without using the analytical definition in Equation 5.7. Thus, the constraint derivative is calculated in its most general form by following its definition in Equation 4.9. This check serves the purposes of validating the derived analytical expression for the constraint derivative in Börjesson et al. (2022).

Again, the central difference is implemented as a quick check, due to the same reasons mentioned above. A perturbation of 5.0×10^{-10} m is used. The results for the constraint derivative in randomly selected elements where energy is being dissipated in substep 18, 19 and 20 is given in Table 5.2. The tendencies seen in Table 5.2 are also valid for the remaining elements wherein energy is being dissipated. The elements are selected randomly, as it is difficult to do it systematically, due to the adaptive floating node method.

Substep 18, 19 and 20 are all located around point (1) in Figure 5.1. Point (1) is where the onset of delamination is happening. It must be noted that the tendencies is not isolated to this point, but valid for other locations on the load-displacement curve when convergence issues occurs.

The step size used is $\Delta\tau = 0.001$. In Table 5.2, three methods are used to calculate the constraint derivative $\{\boldsymbol{h}\}$. Each method is listed below:

- Method 1: An overall central difference is used to calculate the constraint derivative $\{\boldsymbol{h}\}$, according to Equation 4.9
- Method 2: Central difference is used to calculate $\left\{\frac{\partial \Delta Q}{\partial \{\boldsymbol{s}_L\}}\right\}$ and $\left\{\frac{\partial \{\boldsymbol{s}_L\}}{\partial d_j}\right\}$ respectively and the constraint derivative $\{\boldsymbol{h}\}$ is afterwards calculated according to Equation 5.7.
- Method 3: Analytical expressions are used to calcualte $\left\{\frac{\partial \Delta Q}{\partial \{\boldsymbol{s}_L\}}\right\}$ and $\left\{\frac{\partial \{\boldsymbol{s}_L\}}{\partial d_j}\right\}$ and the constraint derivative $\{\boldsymbol{h}\}$ is afterwards calculated according to Equation 5.7.

Random element in substep 18

	$\frac{\partial g}{\partial u_1^+}$	$\frac{\partial g}{\partial v_1^+}$	$\frac{\partial g}{\partial u_1^-}$	$\frac{\partial g}{\partial v_2^-}$	$\frac{\partial g}{\partial u_2^-}$	$\frac{\partial g}{\partial u_2^-}$	$\frac{\partial g}{\partial u_2^+}$	$\frac{\partial g}{\partial v_2^+}$
Method 1:	[0.6448	137.9136	-0.6492	-137.8539	-0.9151	-185.9058	0.9062	186.0373]
Method 2:	[0.6170	131.0154	-0.6170	-131.0154	-0.8803	-179.0993	0.8803	179.0993]
Method 3:	[0.6096	128.5006	-0.6243	-133.5302	-0.8730	-176.5845	0.8876	181.6140]

Random element in substep 20

	$\frac{\partial g}{\partial u_1^+}$	$\frac{\partial g}{\partial v_1^+}$	$\frac{\partial g}{\partial u_1^-}$	$\frac{\partial g}{\partial v_2^-}$	$\frac{\partial g}{\partial u_2^-}$	$\frac{\partial g}{\partial u_2^-}$	$\frac{\partial g}{\partial u_2^+}$	$\frac{\partial g}{\partial v_2^+}$
Method 1:	0.5981	61.9642	-0.5980	-61.9627	-0.4893	-42.7372	0.4893	42.7376
Method 2:	0.5295	55.0691	-0.5295	-55.0691	-0.4074	-35.8421	0.4074	35.8421
Method 3:	0.5290	54.9273	-0.5299	-55.2110	-0.4070	-35.7002	0.4078	35.9840

Random element in substep 19

	$\frac{\partial g}{\partial u_1^+}$	$\frac{\partial g}{\partial v_1^+}$	$\frac{\partial g}{\partial u_1^-}$	$\frac{\partial g}{\partial v_2^-}$	$\frac{\partial g}{\partial u_2^-}$	$\frac{\partial g}{\partial u_2^-}$	$\frac{\partial g}{\partial u_2^+}$	$\frac{\partial g}{\partial v_2^+}$
Method 1:	[0.0006	7.1212	-0.0005	-7.1266	-0.0005	-7.1421	-0.4177	7.1175]
Method 2:	[0.0005	0.2186	-0.0005	-0.2186	-0.0005	-0.2220	0.0005	0.2220]
Method 3:	[0.0005	0.1646	-0.0008	-0.2726	-0.0005	-0.1680	0.0008	0.2760]

Table 5.2. *Comparison of the constraint derivative $\{\boldsymbol{h}\}$ calculated by different methods in randomly selected elements in three substeps.*

Based on Table 5.2 it is clear, that the constraint derivative, $\{\boldsymbol{h}\}$, calculated by all methods in the randomly selected element in substep 18 and 20 are approximately identical. However, for substep 19 method 1 differs alot from the other methods while method 2 and 3 are approximately identical.

Yet another tendency, which is evident from Table 5.2, is that the difference between the methods decreases as the values of the h-vector increases. Both of these tendencies is further found for all the remaining h-vectors calculated in the cohesive elements throughout the model. This may indicate, that the perturbation used in the central difference is not optimal in all elements. However, in order to make this conclusion, it needs to be investigated further.

The arc-length solver is also tested in a configuration, where method 1 is used to calculate the constraint derivative, $\{\boldsymbol{h}\}$. This made the solver more robust, as the step-size, $\Delta\tau$, can now be reduced to lower values than if the constraint derivative is based on the analytical

expressions. A plot of the local energy arc-length solver run with the central difference method and a step size of $\Delta\tau = 0.0001$ is illustrated in Appendix G.

This also indicates that the implementation in the MATLAB Framework, including assembling the constraint derivative and summing up the dissipated energy from each element is done correctly. This part of the implementation has also caused troubles as the constraint derivative, $\{\boldsymbol{h}\}$, is assembled in several steps due to the Floating Node method, which complicates the bookkeeping part.

It is further assessed that the analytical expressions must be partly correct, when it is possible to achieve almost the same results with the central difference and the analytical approach. Based on the random variation between the methods used to calculate the constraint derivative, $\{\boldsymbol{h}\}$, it seems like there are terms which the analytical expression does not account for.

No final conclusion on what causes the convergence issues can be made and further debugging has not been done due to time constraint. Instead a different energy based arc-length method is introduced and implemented in the next section.

5.2 Global Energy Arc-Length Solver

This arc-length solver is implemented as problems were encountered in the implementation of the local energy based arc-length solver in the previous section.

In this section it is first explained how a energy dissipation based path-following constraint, can be formulated in terms of the global displacement and force vector. Finally, this method is implemented in the MATLAB Framework and validated.

The formulation of the constraint is based on the previous mentioned work in in the beginning of this chapter by Verhoosel et al. (2009) and Gutierrez (2004). Also mentioned, this method is referred to as the *global energy arc-length solver*.

A benefit of using this path-following constraint is the fact, that it is based on the global displacement and force vector. This eases the implementation in the MATLAB Framework and thereby reduces the chances for introducing bugs in the code.

The rate of dissipation, G, may be defined as the exerted power, P minus the rate of elastic energy, $\dot{V}$, as done in Equation 5.33.

$$G = P - \dot{V} \tag{5.33}$$

The exerted power, P, is in general defined in terms of the external forces, $\{\boldsymbol{F}_{\text{ext}}\}$, and the nodal velocities, $\{\dot{\boldsymbol{D}}\}$, as in Equation 5.34.

$$P = \{\boldsymbol{F}_{\text{ext}}\}^T\{\dot{\boldsymbol{D}}\} = \lambda\{\hat{\boldsymbol{F}}_{\text{ext}}\}^T\{\dot{\boldsymbol{D}}\} \tag{5.34}$$

The expression for the rate of elastic energy, $\dot{V}$, is problem dependent. In the following a geometrically linear model with damage is assumed as this is consistent with the MATLAB Framework explained in chapter 3. In this case the stored elastic energy is defined in Equation 5.35.

$$V = \frac{1}{2}\int_{\Omega}\{\boldsymbol{\epsilon}\}^T\{\boldsymbol{\sigma}\}\ d\Omega + \frac{1}{2}\int_{\Gamma_{cz}}\{\boldsymbol{s}_L\}^T\{\boldsymbol{t}\}\ d\Gamma_{cz} \tag{5.35}$$

Where the first term is the stored elastic energy in the continuum elements and the second term is the stored elastic energy in the cohesive elements. $\{\boldsymbol{\epsilon}\}$ is the strain, $\{\boldsymbol{\sigma}\}$ is the Cauchy stress, $\{\boldsymbol{s}_L\}$ is the displacement jump vector and $\{\boldsymbol{t}\}$ is the tractions.

Each of the two terms in Equation 5.35 is treated separately and subscript Ω and Γ is used to indicate that the variable is related to the continuum elements or the cohesive elements, respectively. First the term related to the stored elastic energy in the continuum material is considered. In a finite element framework the strain is defined in terms of the strain-displacement matrix, $[\boldsymbol{B}]$, and the nodal displacements, $\{\boldsymbol{d}\}$, as in Equation 5.36.

$$\{\boldsymbol{\epsilon}\} = [\boldsymbol{B}]\{\boldsymbol{d}\}_\Omega \tag{5.36}$$

Inserting Equation 5.36 in the first term in 5.35 yields Equation 5.37

$$\frac{1}{2}\int_\Omega \{\boldsymbol{\epsilon}\}^T\{\boldsymbol{\sigma}\}\, d\Omega = \frac{1}{2}\{\boldsymbol{d}\}_\Omega^T \int_\Omega [\boldsymbol{B}]\{\boldsymbol{\sigma}\}\, d\Omega \tag{5.37}$$

The integral on the right hand side of Equation 5.37 is recognized as the internal forces, $\{\boldsymbol{f}_{\text{int}}\}$, in the system, which means Equation 5.37 simplifies to Equation 5.38

$$\frac{1}{2}\int_\Omega \{\boldsymbol{\epsilon}\}^T\{\boldsymbol{\sigma}\}\, d\Omega = \frac{1}{2}\{\boldsymbol{d}\}_\Omega^T\{\boldsymbol{f}_{\text{int}}\}_\Omega \tag{5.38}$$

Now the second term in Equation 5.35 is considered. The displacement jump vector in the cohesive element is defined in Equation 5.20 and repeated below to ease the reading.

$$\{\boldsymbol{s}_L\} = [\boldsymbol{T}][\bar{\boldsymbol{N}}]\{\boldsymbol{d}\}_\Gamma \tag{5.20}$$

Inserting this in the second term in Equation 5.35 yields Equation 5.39.

$$\frac{1}{2}\int_{\Gamma_{cz}} \{\boldsymbol{s}_L\}^T\{\boldsymbol{t}\}\, d\Gamma_{cz} = \{\boldsymbol{d}\}_\Gamma^T \frac{1}{2}\int_{\Gamma_{cz}} [\boldsymbol{T}]^T[\bar{\boldsymbol{N}}]^T\{\boldsymbol{t}\}\, d\Gamma_{cz} \tag{5.39}$$

According to Turon (2007), the integral in Equation 5.39 is the internal force vector in the cohesive element. Thus, Equation 5.39 reduces to Equation 5.40.

$$\frac{1}{2}\int_{\Gamma_{cz}} \{\boldsymbol{s}_L\}^T\{\boldsymbol{t}\}\, d\Gamma_{cz} = \{\boldsymbol{d}\}_\Gamma^T \frac{1}{2}\{\boldsymbol{f}_{\text{int}}\}_\Gamma \tag{5.40}$$

An expression for the stored elastic energy, V, in the model is determined by combining Equation 5.35, 5.38 and 5.40 and assembling the element displacement vector, $\{\boldsymbol{d}\}$, and internal force vector, $\{\boldsymbol{f}_{\text{int}}\}$ to a global level.

$$V = \frac{1}{2}\{\boldsymbol{D}\}_\Omega^T\{\boldsymbol{F}_{\text{int}}\}_\Omega + \{\boldsymbol{D}\}_\Gamma^T\frac{1}{2}\{\boldsymbol{F}_{\text{int}}\}_\Gamma = \{\boldsymbol{D}\}^T\frac{1}{2}\{\boldsymbol{F}_{\text{int}}\} \tag{5.41}$$

Assuming the system is in equilibrium, i.e. the internal force is in equilibrium with the applied external load, Equation 5.38 is rewritten to Equation 5.42.

$$V = \frac{1}{2}\{\boldsymbol{D}\}^T\lambda\{\hat{\boldsymbol{F}}_{\text{ext}}\} \tag{5.42}$$

By time differentiation of Equation 5.42 the rate of change of the elastic energy, which is needed for the path-following constraint is derived, as in Equation 5.43.

$$\dot{V} = \frac{1}{2}\dot{\lambda}\{\boldsymbol{D}\}^T\{\hat{\boldsymbol{F}}_{\text{ext}}\} + \frac{1}{2}\lambda\{\dot{\boldsymbol{D}}\}^T\{\hat{\boldsymbol{F}}_{\text{ext}}\} \tag{5.43}$$

By combining Equation 5.33, 5.34 and 5.43 the rate of dissipated energy, G, is defined as in Equation 5.44.

$$G = \frac{1}{2}\{\hat{\boldsymbol{F}}_{\text{ext}}\}^T\big(\lambda\{\dot{\boldsymbol{D}}\} - \dot{\lambda}\{\boldsymbol{D}\}\big) \tag{5.44}$$

The corresponding incremental path-following constraint, is found by using a forward Euler discretisation and introducing the step size, $\Delta\tau$, as done in Equation 5.45.

$$g = \frac{1}{2}\{\hat{\boldsymbol{F}}_{\text{ext}}\}^T\big(\lambda^{n-1}\{\Delta\boldsymbol{D}\}_i^n - \Delta\lambda_i^n\{\boldsymbol{D}\}^{n-1}\big) - \Delta\tau \tag{5.45}$$

where λ^{n-1} and $\{\boldsymbol{D}\}^{n-1}$ are the converged load-level parameter and the displacements from the last converged substep, respectively. Thereby, a path following constraint based on the rate of dissipated energy, defined in terms of the global displacement and force vector is derived.

The global energy arc-length solver explained in this section is implemented in the MATLAB Framework by utilizing the same set of equations as in Riks arc-length solver. Similar to the local energy arc-length solver, the displacement and load-level parameter corrections are found by solving the set of equations in Equation 4.8 and 4.9 which are repeated below. Further details on Riks arc-length method is explained in Appendix B.

$$\begin{bmatrix} [\boldsymbol{K}_t]_i^n & -\{\hat{\boldsymbol{F}}_{ext}\} \\ (\{\boldsymbol{h}\}_i^n)^T & w_i^n \end{bmatrix} \begin{Bmatrix} \{\delta\boldsymbol{D}\}_i^n \\ \delta\lambda_i^n \end{Bmatrix} = \begin{Bmatrix} \{\boldsymbol{R}\}_i^n \\ -g_i^n \end{Bmatrix} \tag{4.8}$$

$$[\boldsymbol{K}_t]_i^n = \frac{\partial\{\boldsymbol{F}_{int}\}}{\partial\{\boldsymbol{D}\}} \qquad \{\boldsymbol{h}\}_i^n = \frac{\partial g}{\partial\{\boldsymbol{D}\}} \qquad w_i^n = \frac{\partial g}{\partial\lambda} \tag{4.9}$$

Again, the derivatives of the arc-length constraint with respect to the displacements and the load-level parameter is needed. The derivative of the path-following constraint with respect to the displacements is given in Equation 5.46 while the derivative of the path-following constraint with respect to the load-level parameter is defined in Equation 5.47.

$$\{\boldsymbol{h}\}_i^n = \frac{\partial g}{\partial\{\boldsymbol{D}\}} = \frac{1}{2}\lambda^{n-1}\{\hat{\boldsymbol{F}}_{\text{ext}}\}^T \tag{5.46}$$

$$w_i^n = \frac{\partial g}{\partial\lambda} = -\frac{1}{2}\{\hat{\boldsymbol{F}}_{\text{ext}}\}^T\{\boldsymbol{D}^{n-1}\} \tag{5.47}$$

Equivalent to the local energy arc-length solver, the global arc-length solver needs to be initiated with a solver which does not rely on energy dissipating mechanism. This is explained in more detail in subsection 5.1.1. The solver algorithm for the global energy arc-length solver implemented in the MATLAB Framework is available in Appendix H

5.2.1 Validation of Global Energy Arc-Length Solver

The global energy arc-length solver is validated on the same problem and in the same way as the local energy arc-length solver and the Crisfield geometric arc-length solver. Thus, first solving the problem with the displacement controlled Newton-Raphson in the MATLAB Framework and comparing it with a solution obtained by the global energy arc-length solver. The model used is presented in section 4.2.

The results from each of the test is evident from Figure 5.2. The solution is initialized with the Crisfield field arc-length solver.

To check robustness, the step size, $\Delta\tau$, is varied from a small value of 0.0005 , which yields a fine discretisation of the curve, to a rather large value of 0.01 , which yields a coarse discretisation of the load-displacement curve. During testing of the arc-length solver it was found that if the step size, $\Delta\tau$, is too high, the solver might run into convergence issues. To circumvent this problem, a condition that bisects the step size is implemented in the solver. The bisecting is similar to the bisecting condition done for the Crisfield arc-length solver in section 4.2.

To further check robustness the residual tolerance is varied. A residual of 1×10^{-4} N is used as it yielded relative fast solution times. Further, a residual of 1×10^{-8} N is tested for both the largest and smallest step size, in order to verify that it is possible to obtain solutions with even smaller residuals, if needed. Equivalent, to the validation of the Crisfield solver this is mainly motivated by the fact that bugs in the code have caused convergence problems for some solver settings.

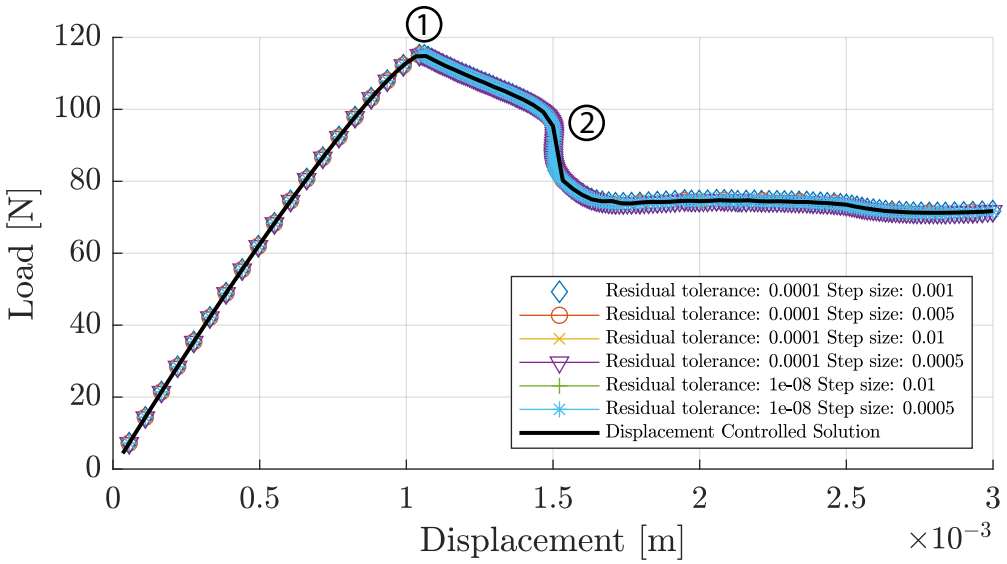


Figure 5.2. *Validation of global energy arc-length solver. The Crisfield arc-length solver is used until point 1.*

From Figure 5.2 it is deemed that all the tests, traces the displacement controlled solution satisfactory. Furthermore the global energy arc-length solver is able to represent the snap-back at point (2). Thus, the same robustness issues related to convergence experienced with the local energy arc-length solver is not experienced with the the global energy arc-length solver.

Literature Study on Toughening Methods

6

As mentioned in section 1.3, the numerical study done by Trabal et al. (2022a) indicates that introducing a patch with increased or decreased onset traction, τ^o, and critical energy release rate, G_c, can promote multiple delaminations, which results in an overall toughening of the structure. In order to realize this a literature study is done exploring the state of the art methods of toughening. Ideally, a literature study on weakening methods should be done as well, but it has proven difficult to find literature about this subject.

It must be noted, that the goal of the literature study is not to choose a best method, but only to explore the possibilities available, which is then used for the experiments, presented in chapter 8 and 9. It can be noted that a good toughening/weakening method is one which is cost effective, easy to manufacture and does not considerably degrade the mechanical, electrical and thermal properties (İnal O. et al., 2022).

In this chapter the principles of the different methods is presented. Figure 6.1 shows an overview of the different methods that has been studied in this project.

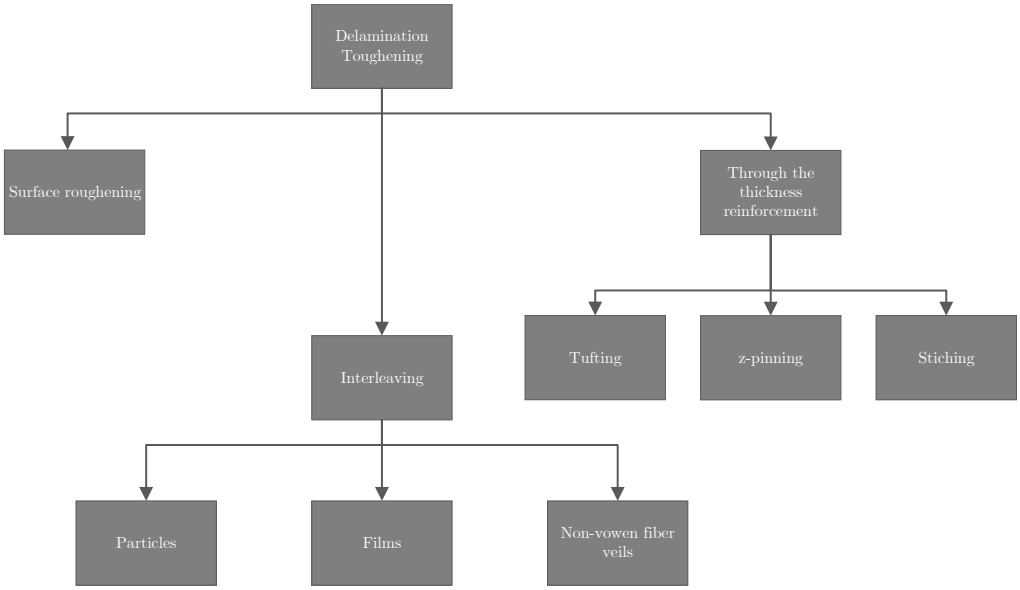


Figure 6.1. *Overview of studied toughening methods.*

6.1 Interleaving

Interleaving methods works by inserting a thin layer of a secondary material in between the layers of the laminate. This is to improve the interlaminar fracture toughness and impact

resistance (Shivakumar and Panduranga, 2013) (Yasaee et al., 2012). The way this can be done is plentiful and a lot of effort in different methods can be found in the literature.

Particles

Interleaving with particles is done by introducing e.g. polymer, nano-silica, or CNT's particles on the surface of the interface, between the layers, by various methods such as sifting, particle filtering and spraying (Shivakumar and Panduranga, 2013) (İnal O. et al., 2022). The main challenge using particles as interleaves is getting an even distribution, and having a good bond between the particles and the rest of the laminate. Uneven distribution can occur for different reasons, such as poor initial distribution, the particles moving with the resin during infusion causing them to agglomerate (İnal O. et al., 2022). Furthermore, even though particles interleaves increases fracture toughness, their thickness is between 20% and 50% of the fiber layer thickness, resulting in increased laminate thickness, thereby reducing the fiber volume fraction in the laminate, which results in reduced in-plane stiffness and strength (Shivakumar and Panduranga, 2013) (İnal O. et al., 2022).

Films

Using a polymer film offers the advantage of a more uniform distribution of the polymer, more rapid moulding, and a cheaper process overall when compared to particle interleaving (Cheng et al., 2019). Furthermore, it is possible to mix different polymers and implement additional phases, such as nanofibers, to achieve multi-scale toughening (Shivakumar and Panduranga, 2013). The downside of using a film compared to particles, is that problems with proper infusion of the resin can occur, due to the film acting as a barrier (İnal O. et al., 2022) (Shivakumar and Panduranga, 2013) (Cheng et al., 2019). Researchers have tried to solve this problem by using porous films or films with holes instead, decreasing the problems with proper infusion (Cheng et al., 2019). Furthermore films has the same problem of reduced in-plane properties as particles due to a thickness increase of the laminate.

Non-vowen fiber veils

Non-vowen fiber veils are mats formed from continuous or discontinuous fibers, which is inserted as an interleave. They can be categorised into two types; microfiber veils and nanofiber veils. The main difference between them is the size of the fibers used, i.e. micro-sized fibers and nano-sized fibers respectively. Both types has high porosity, which allows for good resin flow, and thereby good infusion of the resin. They have proven to significantly improve the interlaminar fracture toughness, while being easy to implement in an existing layup process. Furthermore, they also have the possibility to implement additional phases, so called multi-scale toughening, in order to provide further intralaminar fracture toughness. (İnal O. et al., 2022)

Microfiber non-woven veils consist of randomly distributed micro-sized fibers made from various materials such as termoplastic, carbon, glass, aramid and mixed variations of these. They are mass-producible, cheap and are commercially available in various areal weights. The main downside of using microfiber non-vowen veils is that they add thickness, which results in reduced in-plane stiffness and strength. (İnal O. et al., 2022)

Nanofiber non-woven veils is the latest development in interleaves. They are produced through a process called electrospinning where bulk thermoplastics is spun into continuous fiber, either directly onto a base fiber layer, or onto a support which can then be inserted

in between layers. Its benefits is that they are thin, light, has high porosity and therefore has a minimal, almost negligible, impact on the laminate in terms of weight, thickness, infusion and in-plane stiffness and strength, while increasing the out-of-plane properties. There are still challenges in terms of mass production, and commercially available nanofiber non-vowen veils is therefore limited. (Palazzetti and Zucchelli, 2017) (İnal O. et al., 2022)

6.2 Surface Roughening

The surface roughness has a great effect on the delamination resistance. Generally having a rougher surface improves the delamination resistance. Compared to interleaving which adds extra material in between layers, surface roughening methods does not require an extra layer to be inserted.

Xu et al. (2018) studied the effect of creating a carbon fiber forest (CFF), by pulling out strands of woven carbon fiber mats, by brushing the surface of woven carbon fiber fabrics with a wire brush. The CFF worked as an in situ interleaf, which promoted fiber bridging and improved the G_{Ic}. A maximum loss of in-plane properties similar or lower than other reinforcement methods was observed. Generally, higher levels of brushing done to the surface results in a higher G_{Ic} value, but also a higher loss of in-plane strength. The method is simple, cheap and chemical free and should be possible to incorporate into large scale manufacturing of composites according to the authors.

6.3 Through the Thickness Reinforcement

Through the thickness reinforcement in general aims to improve out-of-plane performance of the laminate, by inserting material transversely to the fiber orientation. Generally, through the thickness reinforcements improves the out-of-plane performance, but decreases the in-plane performance mainly due to the creation of resin pockets and local damage to the fibers. Furthermore, they add costs in the form of specialised equipment and additional steps in the manufacturing process (İnal O. et al., 2022). The three mostly used methods are Z-pinning, stitching and tufting. (Gnaba et al., 2019) (Heß and Himmel, 2010)

Stitching
Stitching works by inserting high strength yarns made of carbon, glass, kevlar etc. through the thickness of the laminate by a needle and tying loops with knots on the other side by a second needle and thread. The loops and tying of knots is one of the main disadvantages to the stitching method, since it causes misalignment and crimping of fibers, due to tension in the yarns. Furthermore, conventional stitching methods requires access to both sides of the laminate, making it harder to apply to mold layups. Advanced stitching technologies has been developed, which only requires access to one side eliminating this disadvantage, but still having to tie knots. (Gnaba et al., 2019)

Z-pinning
Z-pinning involves inserting rigid rods/pins with a thickness of 0.2-1.0 mm through the thickness of the laminate in order to lock the layers together. The pins/rods are made of high strength material such as carbon/epoxy, titanium or steel. (Gnaba et al., 2019)

Tufting

Similar to stitching tufting involves inserting high strength yarns made of carbon, glass, kevlar etc. through the thickness of the laminate. The yarns is inserted through a hollow needle, and the yarns are not tied or interlocked on the other side by a second needle or thread, contrary to stitching, but stays in position solely due to the friction between the fabric and yarn. Therefore only access to one side of the fabric is required. Contrary to stitching tufting do not create tension in the threads themselves and therefore do not have the same problems with misalignment of fibers. (Gnaba et al., 2019)

Parameter Study on Patch Placement Parameters

7

In this chapter a parameter study on the patch placement is done. This is done in an effort to understand which parameters have an effect on the promotion of multiple delaminations. Ultimately conclusions made from the parameter study is used to decide, where to place the patches in the DCB specimens in the experiments presented in chapter 8.

In order to do so, material parameters for the DCB specimen is first approximated. This is followed by a mesh dependency study, with the purpose of ensuring correct modelling of the delamination. Having defined the correct mesh discretisation, it is possible to do the parameter study on the patch parameters.

As mentioned in section 1.3, Trabal et al. (2022a) has mainly investigated onset of multiple delaminations and how this can be used to improve the load bearing capacity. This was done by introducing a patch with augmented interface properties i.e. onset traction τ^o and G_c in the general layup. However, Trabal et al. (2022a) observed that other structural and material parameters also have an effect.

The parameters is investigated by the use of the MATLAB Framework presented in chapter 3. Please note that the loading is different from the experiments presented in chapter 8 and 9, which is loaded with equal and opposite moments. The simulations are done with prescribed displacements, since the developed energy based arc-length solver presented in chapter 4 was not available at the time the parameter study was done. It is expected that the observations made in this chapter also applies for the experiments.

7.1 Specimen Dimensions and Material Properties

Before the simulations can be done dimensions and material properties for the specimens must be specified.

The general layup used for the DCB specimens, from now on referred to as the *virgin laminate*, consists of 16 unidirectional glass fiber noncrimp fabric mats with an areal density of 800 g/m^2, together with PRO-SET INF114-INF212 standard infusion epoxy (PRO-SET inc., 2022). A 0.13 µm thick PTFE film is used to produce an 100 mm initial crack in the mid-plane of the specimens i.e. in the interface between the two center layers. This interface is referred to as the *main interface*. Each layer has a thickness of approximately 0.5 mm when cured. This results in an overall thickness of around 8 mm. The length and width of the specimens is 760 mm and 30 mm, respectively. Patches with augmented interface properties will then be placed in the virgin laminate.

The dimensions and boundary condition of the specimen used in the parameter study is

illustrated in Figure 7.1.

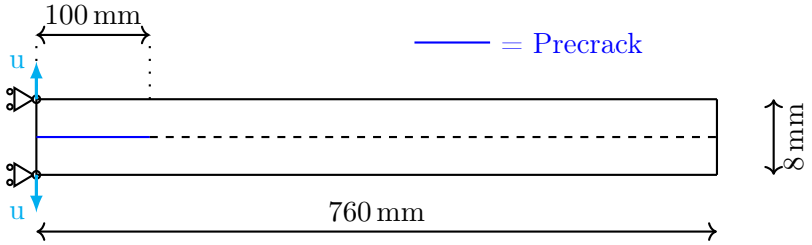


Figure 7.1. *Dimensions and boundary conditions for the DCB specimen used in the preliminary simulations.*

As it is not possible to do material tests before a layup has been manufactured, material properties need to be estimated. The material properties are estimated based on the work by Jensen et al. (2019), Lindgaard and Bak (2019) and Jensen et al. (2018).

Experiments from the aforementioned work has shown critical energy release rates for mode I, G_{Ic}, of around $2000\,\mathrm{J/m^2}$.

No information about critical energy release rate for mode II, G_{IIc}, is available for the material used. G_{IIc} is expected to have a minor influence on the global response due to the mode I loading of the DCB specimens. It is estimated to be four times G_{Ic}, thereby making G_{IIc} $8000\,\mathrm{J/m^2}$.

The onset traction, τ^o, is not as easily obtainable from experimental data as G_c. Numerical studies done by Alfano and Crisfield (2001), Turon et al. (2010) and Lindgaard et al. (2017) indicate that, a too low onset traction for mode I, has a minor effect on the global response during propagation. However, a low onset traction can effect the early loading stage where the model becomes too soft. This can have a negative effect, when predicting onset of delamination and modelling multiple delaminations. On the other hand, using a too low τ^o generally improve convergence by effectively increasing the length of the damage process zone.

Lindgaard et al. (2017) further investigated how the peak load at the early loading stage is affected by the initial crack length and onset traction. They concluded, the onset traction, τ^o, has almost no effect on the peak load for relatively long initial pre-crack. Since a pre-crack of 100 mm is used for the DCB specimens it is assessed that, onset traction, τ^o, has minor influence on the global response. This has been verified in this work for a single delamination but should be investigated for multiple delaminations. Jensen et al. (2018) and Jensen et al. (2019) used a τ_I^o between 30 MPa and 10 MPa. Therefore τ_I^o is set to 30 MPa in this work. τ_{II}^o is then calculated using Equation 7.1 in order to obtain energetically consistent results under mixed-mode conditions (Turon et al., 2010).

$$\tau_{II}^o = \tau_I^o \sqrt{\frac{G_{IIc}}{G_{Ic}}} \tag{7.1}$$

The penalty stiffness, E_p, is determined based on Equation 7.2, which is an expression suggested by Turon et al. (2006).

$$E_p = \frac{\alpha E_{33}}{t} \tag{7.2}$$

E_p is the penalty stiffness, E_{33} is the through-the-thickness Young's modulus, t is the thickness of the layer and α is a parameter much higher than 1 ($\alpha >> 1$). According to Turon et al. (2006) the cohesive surfaces should only simulate fracture, and its contribution to the global deformation should be very small i.e $E_{33} << E_p$. Equation 7.2 ensures that the cohesive contribution to compliance is very small, compared to the surrounding bulk material. Numerical studies done by Turon et al. (2006) showed that $\alpha = 50$ yields an optimum for E_p between convergence and not affecting the global response. If E_p is too high numerical problems such as oscillations and convergence problems can occur.

The material properties used for the simulations are listed in Table 7.1. The elastic properties are the same as Jensen et al. (2019) used in their work. Please note that due to the unidirectional layup used, the elastic properties is transverse orthotropic.

Symbol	Material property	Value
E_{11}	Young's modulus in the fiber direction	21.5 GPa
E_{22}, E_{33}	Youngs's modulus transverse to fiber direction	10.0 GPa
ν_{12}, ν_{13}	Major in-plane Poisson's ratio	0.3
ν_{23}	Major out-of-plane Poisson's ratio	0.07
G_{12}, G_{13}	In-plane shear modulus	4.0 GPa
G_{23}	Out-of-plane shear modulus	2.5 GPa
G_{Ic}	Critical energy release rate Mode I	2000 J/m^2
G_{IIc}	Critical energy release rate Mode II	4000 J/m^2
τ_I^o	Onset traction Mode I	30 MPa
τ_{II}^o	Onset traction Mode II	60 MPa
E_p	Penalty stiffness	5×10^6 N/mm^3

Table 7.1. *Material properties used for the simulations.*

Even though the material properties can not be guarenteed to be the actual values for the specimen, it is assessed to be sufficient for the parameter study. The goal of the parameter study is not to simulate the laminate one-to-one. Instead it is used to study the effect each parameter has on the delamination behaviour.

7.2 Mesh Dependency Study

In order to ensure accurate results, a mesh dependency study based on the overall number of A-FNM elements and user defined ARS-variables, presented in section 3.2, is done in this section for the DCB specimen.

As mentioned in section 3.2, the only ARS-variables that are changed in this work are R_{Zone23} and N_{Zone3}. In order to aid the reader, the definition of the relevant ARS-variable that are investigated is repeated here in short:

- **R_{Zone23}**: Specifies the refinement used in Zone 2 and 3 illustrated in Figure 7.2. Zone 2 is the A-FNM elements, wherein damage is occurring. Zone 3 is the A-FNM elements in front of the crack tip. This is specified as an integer number, which defines the number of sub-elements that are to be used in the longitudinal direction for each of the relevant A-FNM elements.

- $\mathbf{N_{Zone3}}$: Specifies the number of A-FNM elements that are included in Zone 3 illustrated in Figure 7.2. As mentioned, Zone 3 is refined in the same way as Zone 2 i.e. according to the value of R_{Zone23}.

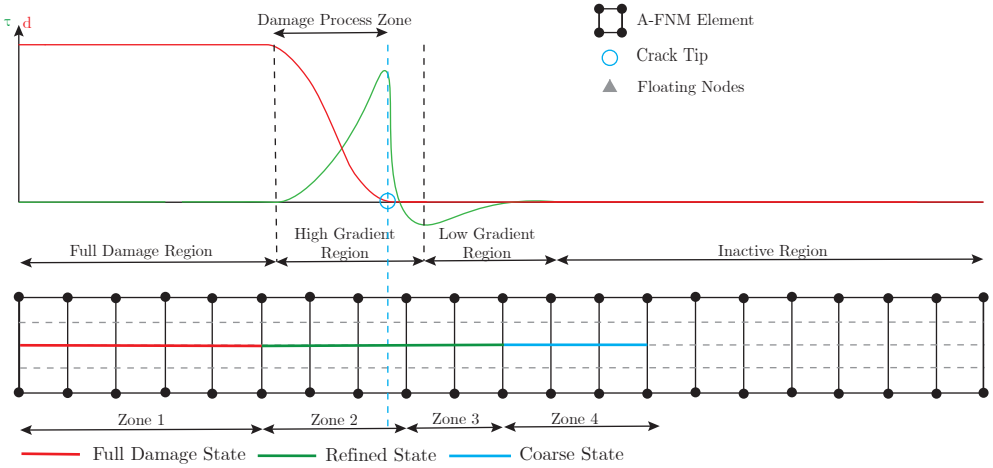


Figure 7.2. *Refinement principle based on the user inputted ARS variables used in the MATLab Framework to control the ARS-algorithm.*

As mentioned in section 3.2, 3-10 cohesive elements are needed in the damage process zone. In order to ensure this, the length of the numerical damage process zone needs to be estimated. It must be noted that the numerical damage process zone length is not the same as the physical damage process zone length (Harper and Hallett, 2008). Several approaches on how to estimate this length can be found in the literature, where it is also commonly referred to as the cohesive zone length. The simplest models for isotropic materials takes the form shown in Equation 7.3.

$$l_{dpz} = ME\frac{G_c}{(\tau^o)^2} \tag{7.3}$$

Where l_{dpz} is the length of the damage process zone, E is Young's modulus and M is a fitting parameter dependant on the model used. The most commonly used models in the literature are Hillerborg et al. (1976)'s model and Falk et al. (2001)'s model, which uses a fitting parameter, M, of 1 and 0.88, respectively. (Turon et al., 2007)

For orthotropic materials Yang and Cox (2005) and Yang et al. (2006) developed modified expressions of Equation 7.3 for both mode I and mode II, as shown in Equation 7.4a and 7.4b, respectively.

$$l_{dpz,I} = E'_I\frac{G_{Ic}}{(\tau^o_I)^2} \tag{7.4a}$$

$$l_{dpz,II} = E'_{II}\frac{G_{IIc}}{(\tau^o_{II})^2} \tag{7.4b}$$

These expressions uses an equivalent elastic modulus, E', for orthotropic materials, which depends on the elastic constants and whether a plane stress or plain strain condition is assumed.

Equation 7.4a and 7.4b is only valid for so called infinite bodies, where the thickness of the body has no influence on the damage process zone length. Most composites is very thin

and the thickness of the body therefore affects the damage process zone length. Due to this, Yang and Cox (2005) developed expressions for thin laminates with centered cracks loaded in mode I or mode II as defined in Equation 7.5a and 7.5b.

$$l_{dpz,thin,I} = \left(E'_I \frac{G_{Ic}}{(\tau_0^o)^2} \right)^{\frac{1}{4}} t^{\frac{3}{4}} \tag{7.5a}$$

$$l_{dpz,thin,II} = \sqrt{\left(E'_{II,thin} \frac{G_{IIc}}{(\tau_{II}^o)^2} \right) t} \tag{7.5b}$$

Where t is half the thickness of the laminate, and $E'_{II,thin}$ is a mode II equivalent young's modulus.

Based on a numerical study, Harper and Hallett (2008) showed that Equation 7.4a, 7.4b, 7.5a and 7.5b generally over predicts the numerical damage process zone length. They suggested that a scaling factor of 0.5 should be applied to the minimum values of the equations, when using a bi-linear traction separation law under mixed-mode conditions.

$$l_{dpz,estimated} = 0.5(\min(\text{Equation 7.4a, 7.4b, 7.5a, 7.5b})) \tag{7.6}$$

As mentioned in section 3.1 the traction-separation law used in the MATLAB Framework is a bi-linear traction-separation law. Equation 7.6 will therefore be used to estimate the length of the numerical damage process zone.

Having defined the expression for the estimated length of the damage process zone, $l_{dpz,est}$, it is possible to determine the maximum element length for each element in the damage processs zone, $l_{ele,max}$, by using Equation 7.7

$$l_{ele,max} = \frac{l_{dpz,estimated}}{N_{ele,DPZ}} \tag{7.7}$$

Where $N_{ele,DPZ}$ is the desired number of elements in the damage process zone.

Using the material properties and specimen dimensions presented in section 7.1, Equation 7.6 yields a estimated damage process zone length, $l_{dpz,estimated}$, of 2.89 mm.

As mentioned in section 3.2, the number of elements in the damage process zone, $N_{ele,dpz}$, has to be between 3 and 10. Setting $N_{ele,DPZ}$ to 10 in Equation 7.7, yields a maximum length of each element in the damage process zone, $l_{ele,max}$, of 0.289 mm. This value is used as a starting point for the mesh dependency study.

All the models done in the mesh dependency study is done with the model in Figure 7.1 and a prescribed displacement, u, of 30 mm.

7.2.1 Number of A-FNM Elements

Trabal et al. (2022a) reports that the discretisation of the length of the A-FNM elements can have an influence on the results. This is due to the A-FNM elements not being able to represent the deformation state. This is mainly due to bending for the DCB specimen. Therefore a convergence study on this is done in order to ensure accurate results.

In order to rule out all other variables in the study of the A-FNM element size, L_{A-FNM}, the element size in the damage process zone, $l_{max,DPZ}$, is kept constant at 0.289 mm.

In practice, $l_{max,DPZ}$ is kept at a constant value by adjusting the ARS-variable R_{Zone23} according to Equation 7.8.

$$R_{Zone23} = \frac{L_{A-FNM}}{l_{max,DPZ}} \tag{7.8}$$

Furthermore, the total length of zone 2 and 3 combined is kept constant, which results in the total number of sub elements being constant. This is done by adjusting the ARS-variable N_{Zone3} according to Equation 7.9.

$$N_{Zone3} = \frac{L_{A-FNM,max}}{L_{A-FNM}} - 1 \tag{7.9}$$

Where $L_{A-FNM,max}$ is the maximum A-FNM element length investigated, i.e. 40 mm.

An overview of the parameters used for the convergence study on the length of the A-FNM elements is shown in Table 7.2, with the size of the smallest and largest mesh illustrated in Figure 7.3.

$\mathbf{N_{A\text{-}FNM}}$	$\mathbf{L_{A\text{-}FNM}}$	$\mathbf{R_{Zone23}}$	$\mathbf{N_{Zone3}}$
19	40 mm	138	0
38	20 mm	70	1
76	10 mm	35	3
152	5 mm	17	7
228	3.3 mm	12	11
304	2.5 mm	9	15

Table 7.2. *AFN-element and ARS-variables used for the convergence study on the number of A-FNM elements.*

(a) *A-FNM element size, L_{A-FNM} = 40 mm.*

(b) *A-FNM element size, L_{A-FNM} = 2.5 mm.*

Figure 7.3. *Illustration of the smallest and largest A-FNM element size used for the convergence study on A-FNM element length. The pre-crack is marked with a blue line*

Figure 7.4 illustrates the load-displacement curves for the different A-FNM element lengths used. It is evident from the figure, that the length of the A-FNM elements does not affect the outcome of the simulation, except in the case of an A-FNM element length of 40 mm. This change is due to the pre-crack not being placed at the same place as the rest of simulations as illustrated in Figure 7.3. The MATLAB Framework can only place pre-cracks in whole A-FNM elements and will do a extension/reduction of the pre-crack to the nearest whole A-FNM element. In the case of an L_{A-FNM} of 40 mm, the pre-crack therefore ends up being placed 120 mm, instead of the intended 100 mm.

As there is no obvious difference in the load-displacement curve for the different A-FNM element lengths, L_{A-FNM}, except for the case discussed above, a L_{A-FNM} of 5 mm is chosen for the remaining simulations in this chapter. The size is chosen as it is a good compromise, between the total number of coarse A-FNM elements, N_{A-FNM} and the refinement in Zone 2 and 3, R_{Zone23}. Additionally, the L_{A-FNM} of 5 mm makes it possible to place patches and pre-cracks at 5 mm intervals.

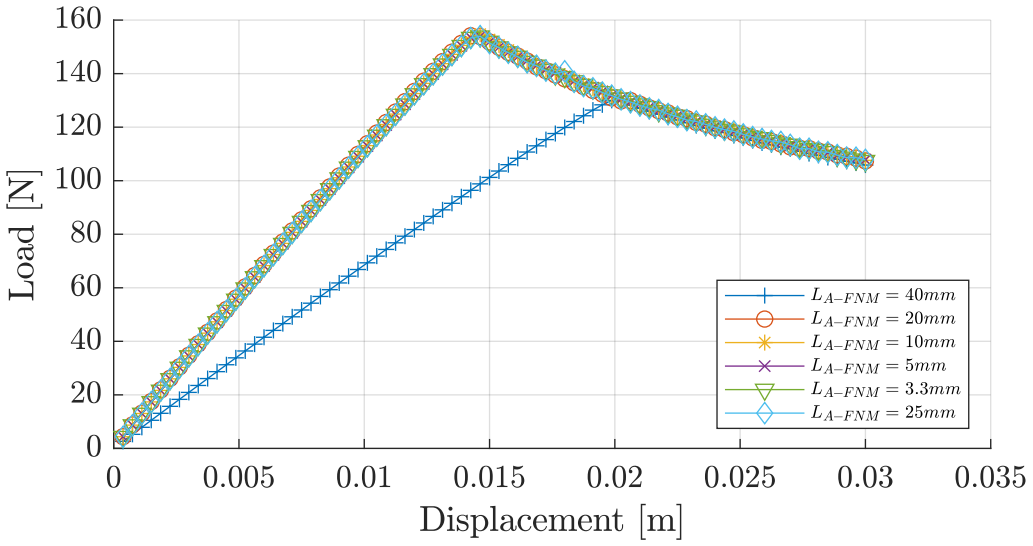


Figure 7.4. *Load displacement equilibrium curve for different A-FNM element lengths.*

7.2.2 Number of sub-elements

As mentioned in section 3.2 3-10 elements in the damage process zone yields a good compromise between convergence and accurate results. Additionally the remaining part of the high gradient region must be discretised with equally fine elements.

In order to check whether this is the case the mode I traction, τ_I, profile is investigated. Afterwards, a study on the number of subelements elements is done in order to check mesh dependency. This is all done on the aforementioned DCB-model with an A-FNM element length of 5 mm.

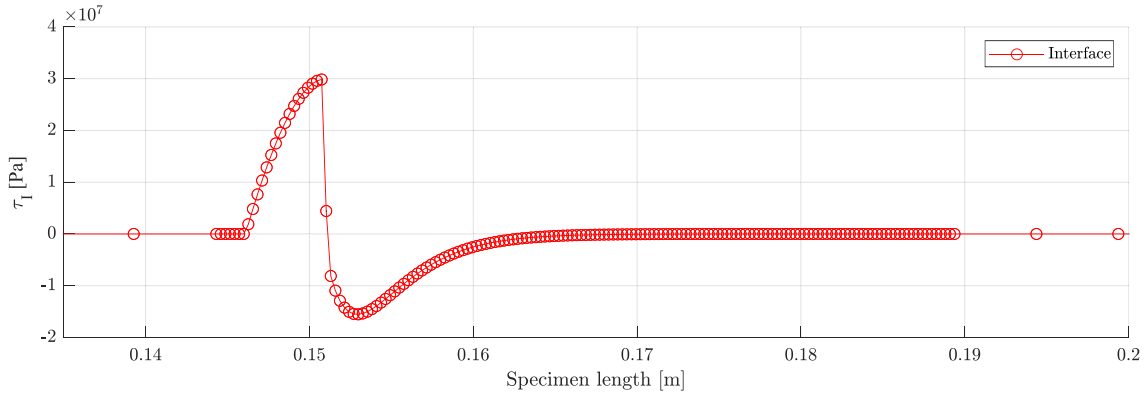


Figure 7.5. *Mode I traction profile around the crack tip, at the full displacement of 30mm. Each circle represents a node.*

Figure 7.5 illustrates the mode I traction profile near the crack tip at the fully applied displacement of 30 mm. As it is evident from Figure 7.5 the length of the damage process zone is about 4.7 mm. This length is longer than the estimated 2.89 mm from Equation 7.6. Consequently 16 elements are present in the damage process zone, which is more than enough to model the delamination correctly. Furthermore, the length of zone 3, defined in Figure 7.2, seems to be plenty long to model the high gradient region, which is about 15 mm, correctly.

In order to reduce the number of subelements, the amount of A-FNM elements in Zone 3 is reduced by adjusting the ARS-variable N_{Zone3} to 1. The reduction is done in order to improve the computational efficiency.

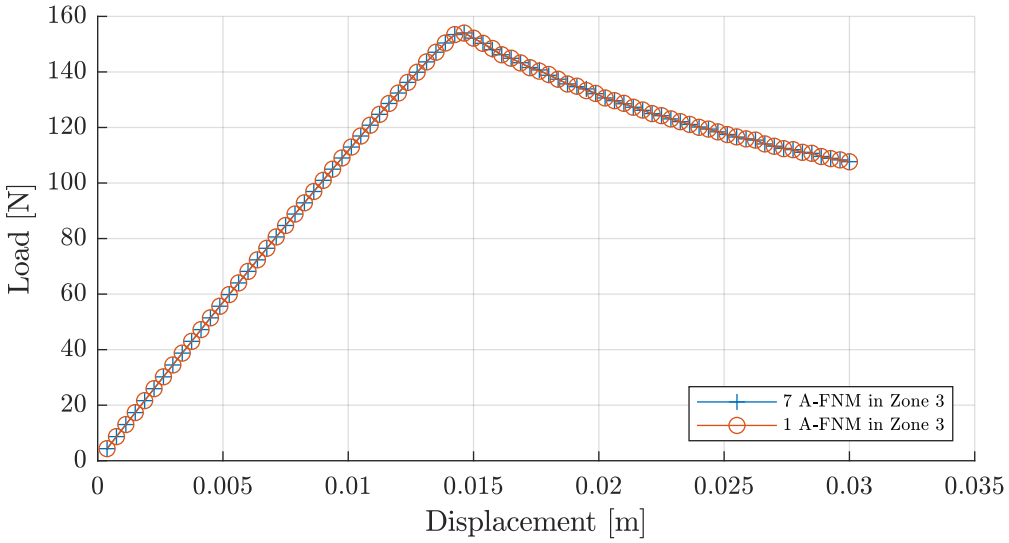


Figure 7.6. *Load displacement equilibrium curve for two different values of N_{Zone3}.*

Figure 7.6 illustrates the load-displacement equilibrium curves for 1 and 7 A-FNM elements included in zone 3. From Figure 7.6 it is observed that the global response is the same. Figure 7.7 illustrates the mode I traction profile around the crack tip for both cases. This also illustrates that 1 A-FNM element in zone 3 is sufficient to capture the high gradient zone and thereby model the delamination correctly.

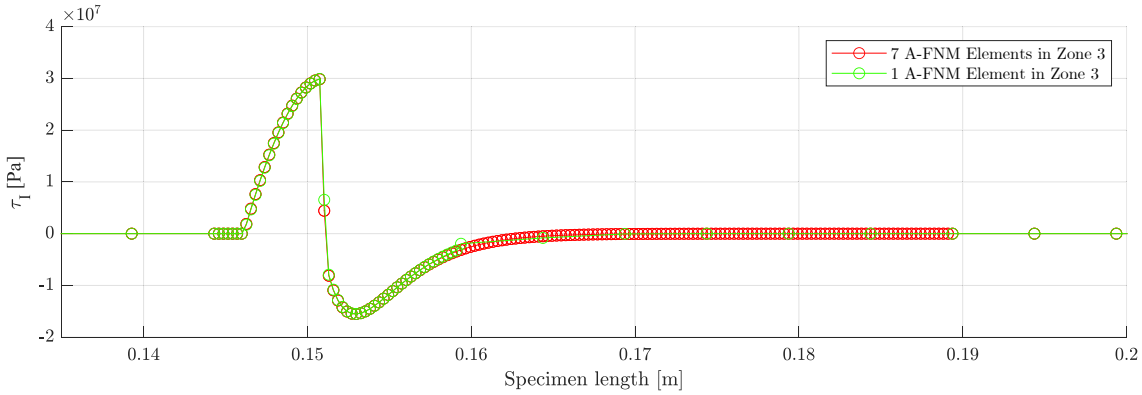


Figure 7.7. *Mode I traction profile around the crack tip, at the full displacement of 30mm for two different values of N_{Zone3}.*

In order to investigate the influence of the number of sub-elements in the damage process zone, a convergence study on this is done. This is done by adjusting the ARS-variable R_{Zone23}.

Figure 7.8 illustrates the load displacement equilibrium curves for the different values of R_{Zone23}. From Figure 7.8 it is observed that the peak load is increased, for the low values of 2 and 4. Furthermore, the response is oscillating once past the peak load. Both these behaviours is a sign that not enough elements are present in the damage process zone.

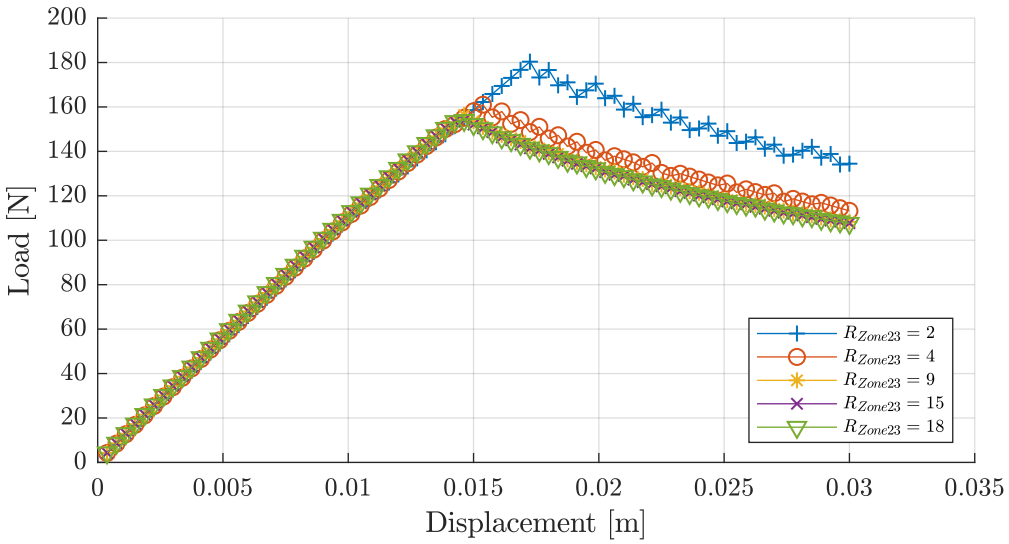


Figure 7.8. *Global response for different refinement values for R_{Zone23}.*

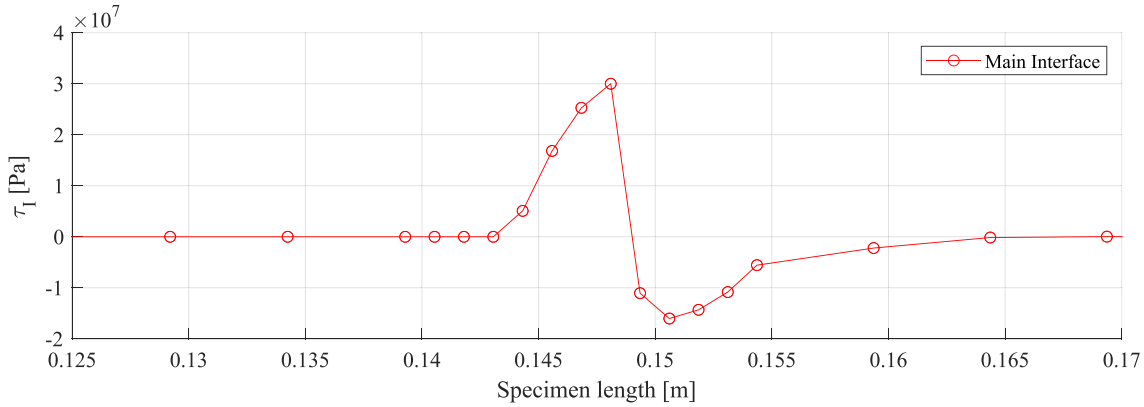


Figure 7.9. *Mode I traction profile near the crack tip at the 30 mm displacement values for a R_{Zone23} value of 4. The circles represents a node.*

Figure 7.9 illustrates the mode I traction profile near the crack tip at the fully applied displacement of 30 mm for a R_{Zone23} value of 4. Figure 7.9 shows that 4 sub-elements are present in the high traction zone. According to Trabal et al. (2022b) this should be enough elements to model the delamination correctly. As mentioned the global response indicates, that this is not the case, and therefore more than 4 elements is needed in the damage process zone.

Based on the convergence study on the number of sub-elements in the damage process zone, the ARS-variable R_{Zone23} is set to 25 for the remaining simulations in this chapter. This relative high value is used to account for a shorter damage process zones, once the toughening patches are introduced. The number of elements present in the damage process zone is checked for all the simulations done in the parameter study in order to ensure correct modelling of the delamination.

7.3 Parameter Study

In this section parameters related to the placement and material of the patches is studied.

The parameters that will be investigated in the parameter study includes:

- Distance between pre-crack and patch
- Patch length
- Interface placement of the weakening patch

The reason for investigating each parameter will be explained throughout this section.

The ARS-variables and number of A-FNM elements used for the parameter study, based on the convergence study in the previous section, is presented in Table 7.3

$L_{A\text{-}FNM}$	$N_{A\text{-}FNM}$	R_{Zone23}	N_{Zone3}
5	152 mm	25	1

Table 7.3. *AFN-element and ARS-variables used for the parameter study.*

7.3.1 Augmented Patch Parameters

A study is done, to check whether multiple delaminations will occur, when a patch with augmented onset traction τ^0 and critical energy release rate G_c, is placed in the virgin laminate.

A weakening patch and two toughening patches is studied separately. The weakening patch is placed in the interface just below the main interface i.e. at a distance of 0.5 mm. This interface is referred to as the secondary interface. The weakening patch has the following interface properties:

- $\tau^o = 0$ and $G_c = 0$ of for both mode I and II

Thereby the weakening patch cannot transfer any loads and essentially acts similar to placing a PTFE film in the secondary interface. This is chosen since this is the way weakening patches will be introduced in the experiments.

The toughening patches are placed in the main interface and has the following properties:

- $2 \cdot \tau^o$ and $2 \cdot G_c$ of the virgin laminate for both mode I and II
- $1.25 \cdot \tau^o$ and $1.25 \cdot G_c$ of the virgin laminate for both mode I and II

The interface properties of the toughening patches is unknown before the experiments are done. Instead above mentioned values for the toughening patches is chosen in order to get an rough idea of what the critical energy release rate, G_c and onset traction, τ^o should be in order to promote multiple delaminations.

The patches is placed at a distance of 50 mm from the pre-crack and has a length of 50 mm. The model used in the study is illustrated in Figure 7.10. The prescribed displacement, u, is 100 mm. Only the main interface and the secondary interface are included in the model, in order to improve computational efficiency.

Figure 7.11 illustrates the load-displacement curve of the DCB-specimen with a toughening patch, weakening patch and without a patch. Figure 7.12 illustrates the deformation of the DCB-specimen at the four points of interest marked in Figure 7.10. The four points of interest marks the following:

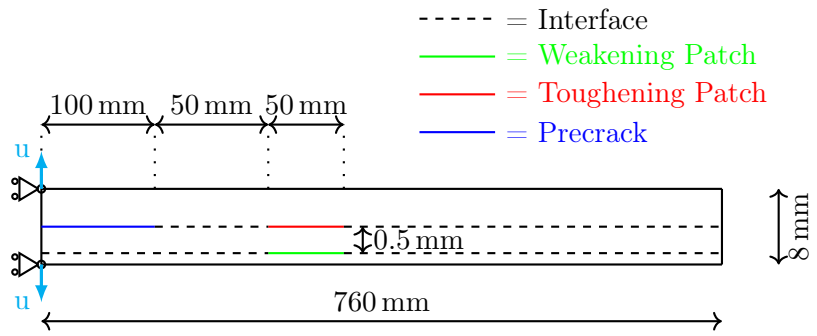


Figure 7.10. *Dimensions and boundary conditions for the DCB specimen used in the study of augmented patch parameters.*

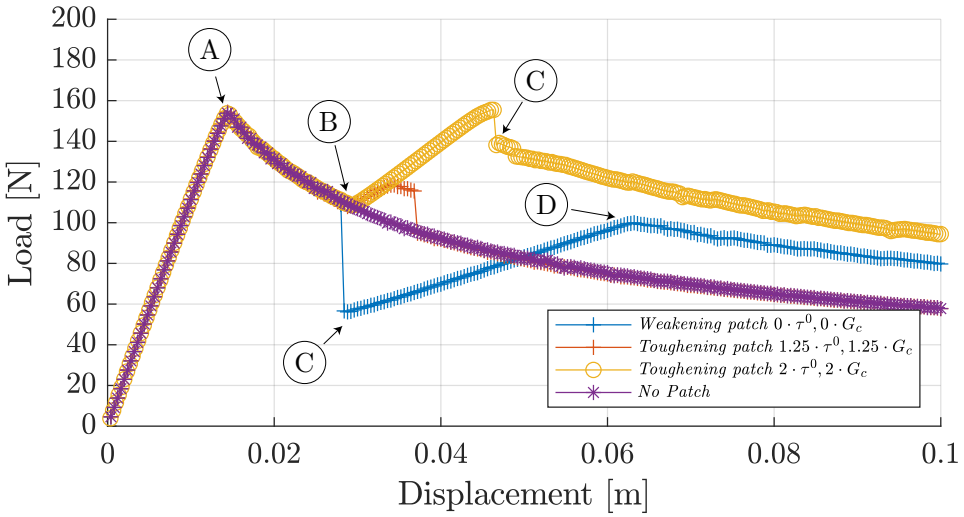


Figure 7.11. *Load-displacement curve of the DCB-specimen with a weakening patch, a toughening patch and without a patch.*

(A) **Both patch types:** Delamination is onset from the pre-crack in the main interface and starts propagating as a single delamination.

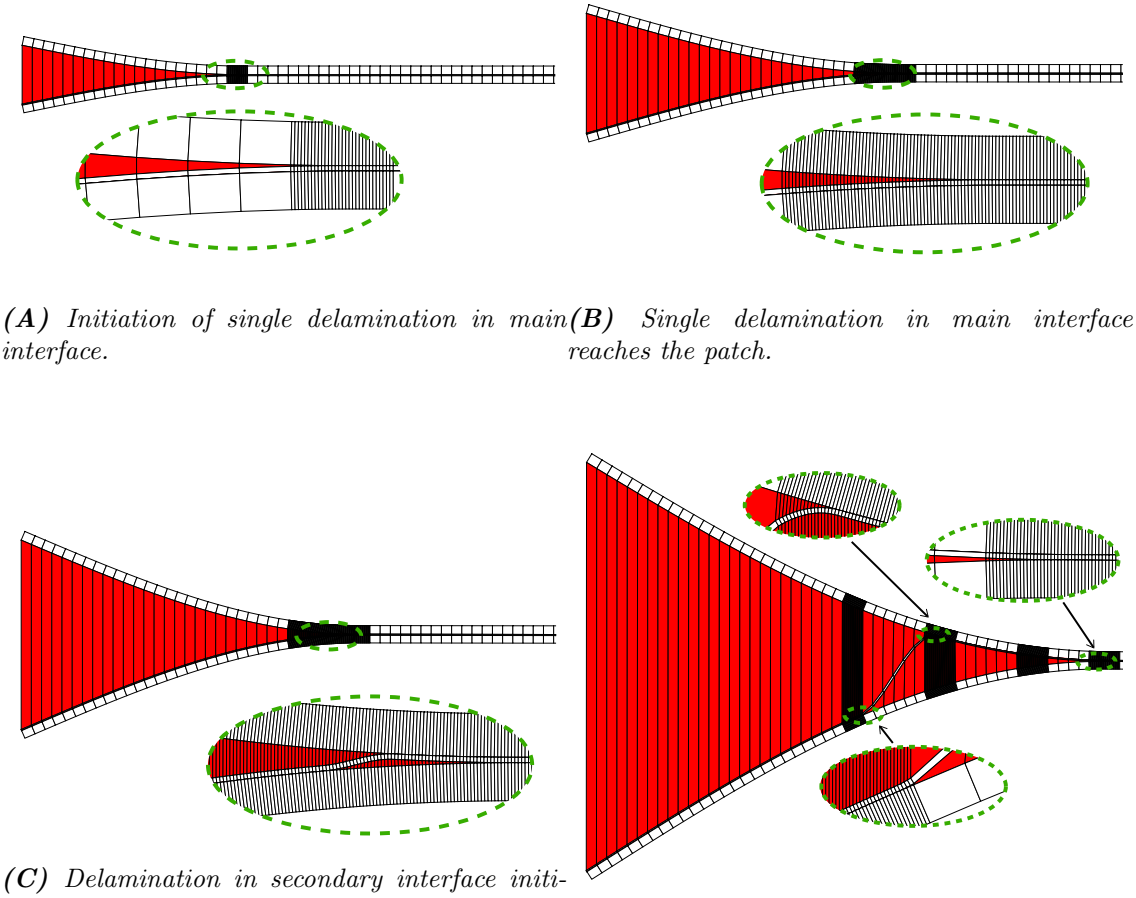
(B) **Toughening patches:** The single delamination in the main interface reaches the toughening patch. This leads to an initial toughening of the specimen during propagation of the single delamination in the main interface, until the peak load is reached.

Weakening patch: The single delamination in the main interface is aligned with the weakening patch.

(C) **Toughening patch:** Delamination in the secondary interface is onset causing a drop in the load. Multiple delamination fronts are present at this stage and starts propagating. The delamination in the toughened main interface propagates slower than the delamination in the secondary interface. This is due to the toughening patch, making it harder for the delamination in the main interface to propagate.

Weakening patch: Delamination in the secondary interface is onset causing a drop in the load. The entire weakening patch opens as soon as the delamination in the secondary interface is onset, hence the large load drop. This is followed by a second load increase. None of the delaminations are propagating.

(D) **Weakening patch:** Multiple delaminations starts propagating.



(A) *Initiation of single delamination in main interface.*

(B) *Single delamination in main interface reaches the patch.*

(C) *Delamination in secondary interface initiates.*

(D) *Propagation of the multiple delaminations.*

Figure 7.12. *DCB-specimen deformation at the four points marked in Figure 7.11.*

From Figure 7.10 it is observed that multiple delaminations are promoted for both the weakening and the toughening patch with $2 \cdot \tau^o$ and G_c. The multiple delaminations leads to a toughening of the specimen, when compared to the virgin laminate without a patch. The toughening patches leads to a higher toughening than the weakening patches. For the toughening patch with $2 \cdot \tau^o$ and G_c an initial toughening is present, but eventually drops of and comes to the same level as a virgin laminate without a patch. This indicates that multiple delaminations has not been promoted. Thereby it indicates that a toughening patch with interface values of around twice the virgin laminate is needed in order to promote multiple delaminations.

7.3.2 Length of Patch

The length of the patch used has an effect on whether multiple delaminations is promoted or not, according to Trabal et al. (2022a). If the delamination in the main interface is not fully arrested, a longer toughening patch increases the probability of onsetting multiple delaminations. Similarly, if a weakening patch is used in the secondary interface

a longer weakening patch increases the probability of onsetting and propagating multiple delaminations. In both cases, the patch needs to be sufficiently long to ensure a full damage process zone, in the secondary interface, is developed with high enough tractions to onset multiple delaminations. (Trabal et al., 2022a)

Three different patch lengths, l_{patch}, is studied; 10 mm, 25 mm and 50 mm. All the patches is placed at a distance of 50 mm from the pre-crack. Only toughening patches is studied. The rest of the settings are the same as described in the previous sections. The model used is illustrated in Figure 7.13.

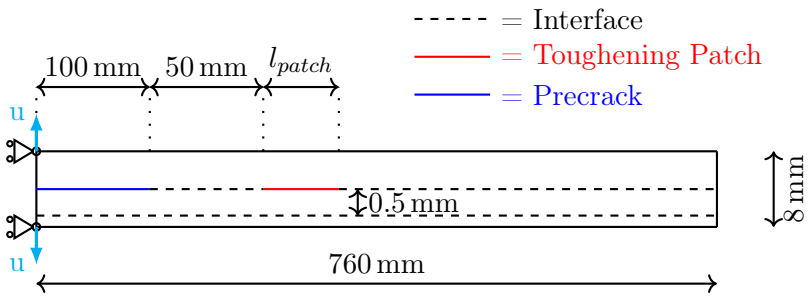


Figure 7.13. *Dimensions and boundary conditions for the DCB specimen used in the study of* l_{patch}.

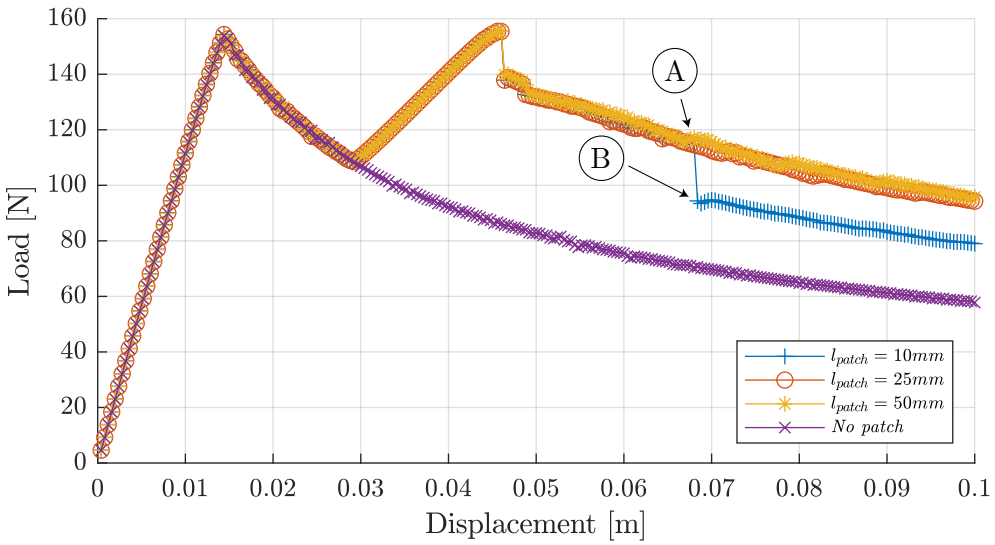
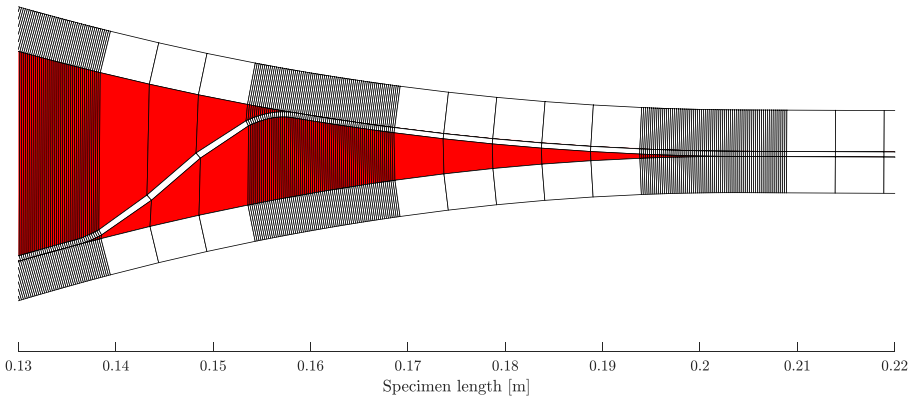
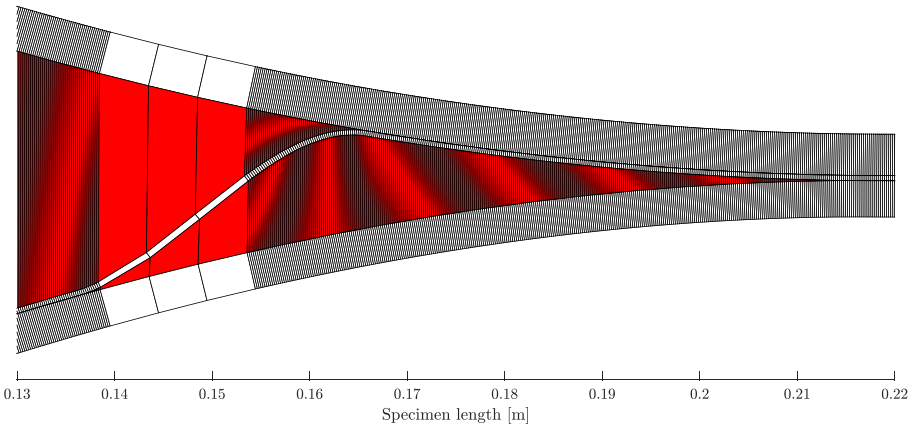


Figure 7.14. *Load-displacement equilibrium curve of the DCB-specimen with different lengths of the toughening patch,* l_{patch}*, and without a patch.*

Figure 7.14 illustrates the load-displacement curve of the DCB-specimen with different lengths of the toughening patch, l_{patch}. The general behaviour is the same as described in subsection 7.3.1 and identical for all l_{patch}, except for the patch with a l_{patch} value of 10 mm. Aforementioned patch length experiences a load drop at 0.07 mm displacement. Figure 7.15 illustrates the deformation at point (A) and (B) marked in Figure 7.14. From aforementioned figures it is evident, that the load drop is due to the delamination in the main interface propagating out of the toughening patch at this point. This causes a sudden propagation both in the main and secondary interface.



(A) *Deformation just before the delamination in the main interface propagates out of the toughening patch.*



(B) *Deformation just after the delamination in the main interface propagates out of the toughening patch.*

Figure 7.15. *Deformation of the DCB specimen with* $l_{patch} = 10\,\text{mm}$ *at point A and B marked in Figure 7.14*

Conclusions made from the study on the length of the patch, is that multiple delaminations is promoted even with a short patch, thereby toughening the specimen. None of the lengths fully arrests the delamination in the main interface. If l_{patch} is too short the delamination in the main interface can propagate out of the toughening patch causing a sudden load drop. Furthermore, since the interface properties of the materials used as toughening patches is unknown a longer patch might increase the probability of promoting multiple delaminations. This needs to be studied further in order to finally conclude this.

7.3.3 Interface Placement of the Weakening Patch

According to Trabal et al. (2022a), a lower distance between the main and secondary interface increases the probability of onsetting and propagating multiple delaminations.

As mentioned in section 7.1 the layup used for the specimens consists of 16 layers which is approximately 0.5 mm thick. Therefore the distance between interfaces is predetermined to 0.5 mm. Instead the effect of the distance between interfaces observed by Trabal et al.

(2022a) can relate to which interface the weakening patch is placed in. A weakening patch placed close to the main interface increases the likelihood of multiple delaminations. The downside of placing it too close to the main interface is that the main and secondary interface might collapse into one interface during the experiments. The interfaces in the specimens in the experiments is not perfectly separated as they are in the simulations, due to mechanisms like fiber bridging, fiber pullout and crack jumping. Therefore a study on whether multiple delaminations is onset, if the weakening patch is placed in different interfaces is done in the following.

Four different interface distances, d_{weak}, is investigated; 0.5 mm, 1 mm, 1.5 mm and 2 mm. The distances corresponds to placing the patch in the first, second, third and fourth interface from the main interface, respectively. The model used is illustrated in Figure 7.16 Only the main interface and the interface where the patch is placed is simulated, due to computational efficiency. The weakening patches are all 50 mm long and placed at distance of 50 mm from the pre-crack. The rest of the settings are the same as described in subsection 7.3.1.

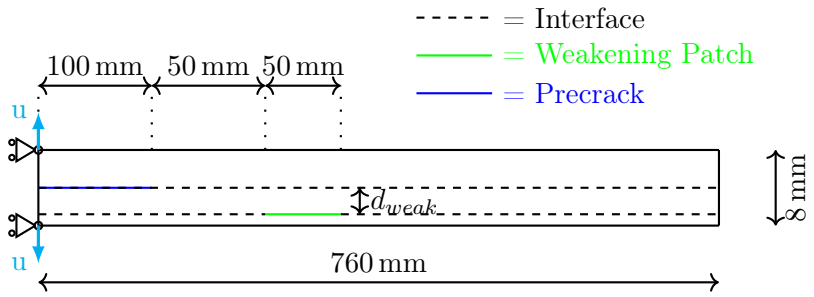


Figure 7.16. *Dimensions and boundary conditions for the DCB specimen used in the study of* d_{weak}.

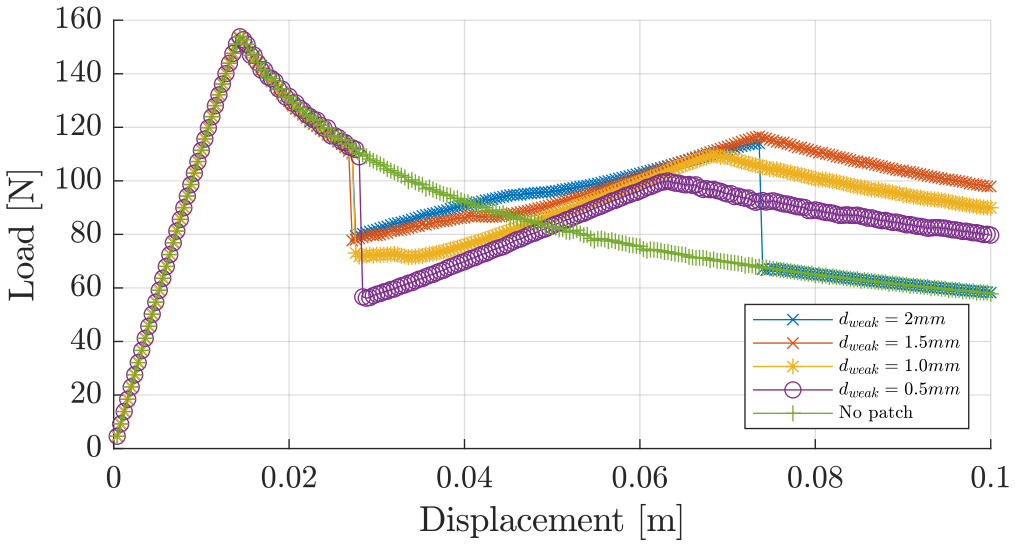


Figure 7.17. *Load-displacement equilibrium curve of the DCB-specimen with different interface distance,* d_{weak}*, and without a patch.*

Figure 7.17 illustrates the load-displacement curve for the simulations. The general behaviour is the same as described in subsection 7.3.1. From Figure 7.17 it is evident that

multiple delaminations is onset and starts propagating for all patch placements except for the patch distance of 2 mm. At this distance the load initially drops due to the full weakening patch being reached. This is followed by an initial toughening until a second load drop happens. The load drops to the same level as a specimen without a patch indicating that no toughening, due to multiple delaminations, is achieved. Thus, it is possible to place the weakening patch at a distance where multiple delaminations is not onset and starts propagating.

7.4 Summary

To summarise, the observations made from the parameter study are the following:

- Multiple delaminations can be onset for a weakening patch with interface properties equal to 0.
- Multiple delaminations can be onset for a toughening patch with twice the virgin laminate interface properties.
- Multiple delaminations are onset even if a short toughening patch is used. This has only been studied for a toughening patch with interface properties twice the virgin laminate.
- A too short toughening patch causes a sudden load drop if the delamination propagates out of the patch.
- Multiple delaminations are promoted if the weakening patch is placed in one of the three secondary interfaces nearest to the main interface.

Based on the observations, DCB specimens with toughening and weakening patches are manufactured and tested in the next chapter.

Experimental Setup and Test Specimens

8

The experimental part of this work is explained in the following chapter. The experimental test are done to determine whether multiple delaminations is onset in the DCB-specimen and whether it has a toughening effect. In order to determine this, the fracture resistance and crack mouth opening displacement (CMOD) is determined in the tests of the DCB-specimens. The chapter is organised in the following way. First the manufacturing, test setup and dataprocessing is explained. Then two preliminary types of DCB specimens are designed, manufactured and tested. This leads to the development of a virtual clip gauge. Finally the results from six different types of DCB specimens are presented.

8.1 Manufacturing Process

As mentioned in section 7.1, the general layup used for the DCB specimens consists of 16 unidirectional glass fiber noncrimp fabric mats with an areal density of 800 g/m2, infused with PRO-SET INF114-INF212 standard infusion epoxy (PRO-SET inc., 2022). The layup is symmetric around the midplane, with the UD fibers facing the midplane. A 0.13 µm thick PTFE film is used to produce an 100 mm pre-crack in the midplane. Each layer has an approximately thickness of 0.5 mm when cured, resulting in an overall thickness of around 8 mm. Patches are then introduced in the general layup in order check their ability to promote multiple delaminations.

The specimens is manufactured by using a vacuum assisted resin transfer molding process (VARTM) and cured according to the specification from the resin manufacturer. An image from the infusion process is shown in Figure 8.1.

In order to manufacture the specimens a single plate is first manufactured with a size of 780 mm by 465 mm. The specimens is then cut from the plate to a size of 740 mm by 30 mm using a CNC-mill with a end mill designed for glass fiber composites. An existing MATLAB script for generating tool paths for the CNC-mill is modified and used to machine the specimens from the plate. The reason to use the CNC-mill is to automate the cutting of the DCB specimens, as a high number of specimens are manufactured in this work.

An image from the machining process is shown in Figure 8.2 After cutting the DCB specimens, holes are drilled and taped in order to attach a mounting bracket to each arm of the DCB specimen. Furthermore, two pins are attached to the DCB specimen in order to mount a clip gauge. These pins are located on one side of the specimen, at the tip of the initial crack. The final dimensions of the specimens are measured with a caliper, for use in the data processing.

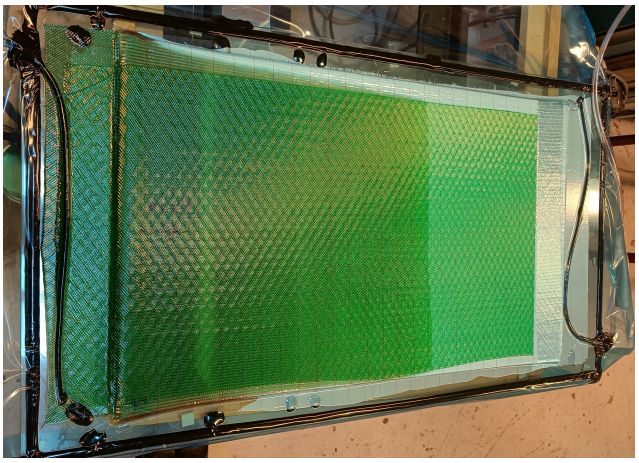


Figure 8.1. *Infusion process.*

Figure 8.2. *Composite plate machined in the CNC machine.*

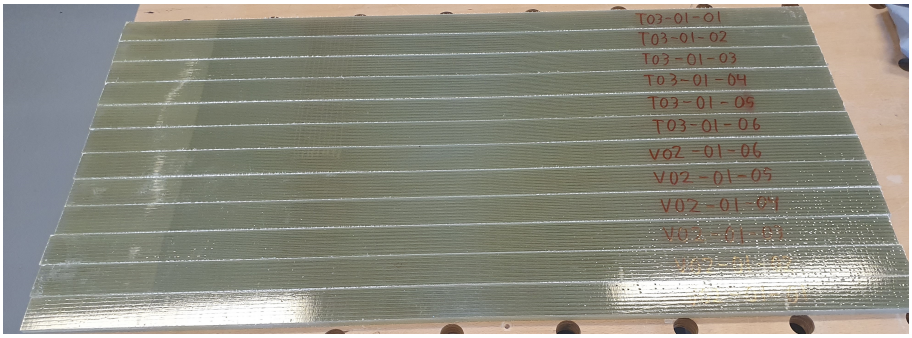


Figure 8.3. *Naming of DCB specimens.*

The DCB specimens are named and numbered based on their layup and position in the cured composite plate e.g. *V01-01* is the first virgin specimen from the first virgin layup. W and T will be used for specimens with a weakening or toughening patch respectively. This is shown in Figure 8.3.

8.2 Test Setup

The DCB specimens are tested using the test fixture, shown in Figure 8.4, which applies pure even or uneven bending moments to the DCB specimen. The test fixture is actuated by a standard tensile testing machine and converts a force loading into bending moments using a wire pulley system. By using a standard tensile testing machine as actuation, the existing high precision motion and data acquisition from the tensile testing machine can be utilised. (Lindgaard and Bak, 2019)

The standard test method for delamination testing of DCB specimens is done according to ASTM D5528-13 and ISO 15024. For mode I testing these methods use wedge loading, where a DCB specimen is mounted with hinges and loaded. Aforementioned methods requires that the crack length is recorded during testing, which can be challenging. One of the benefits of applying bending moments to the DCB specimen, is that the crack length does not need to be monitored (Sørensen et al., 2006).

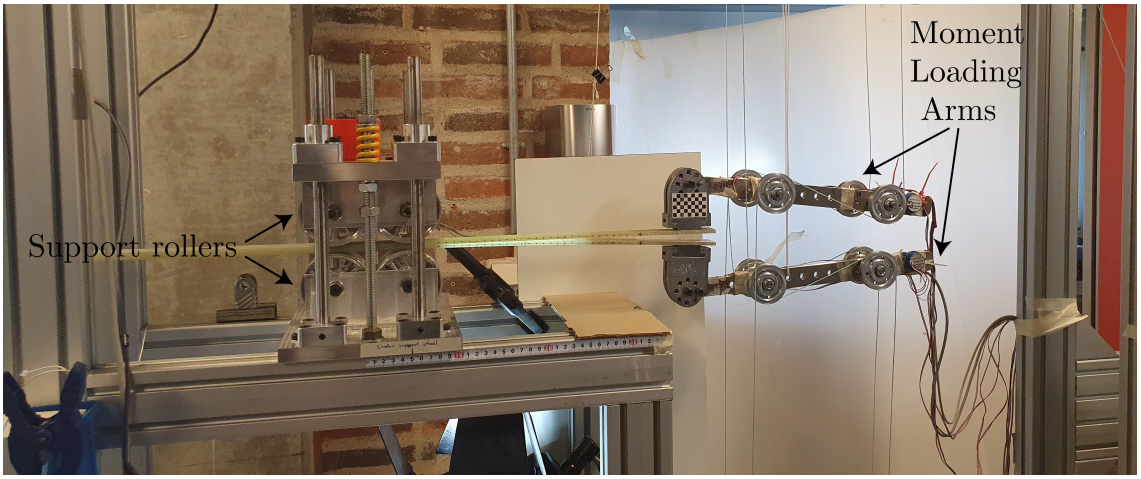


Figure 8.4. *Overview of test fixture used for tests of the DCB specimens.*

The procedure of testing is as follows:

- Attach a mounting bracket to each arm of the DCB specimen.
- Position the DCB specimen in the support rollers.
- Attach the moment loading arms to the specimen.
- Attach clip gauge to the specimen.

Data from the clip gauge is logged together with the force applied by the tensile test machine and the angle of each loading arm. The angle is measured using inclinometers, mounted on the loading arms.

8.3 Experimental Data Processing

The energy release rate needs to be determined, in order to characterise the delamination of the DCB specimens. The energy release is determined based on the J-integral shown in Equation 8.1.

$$G = J = \frac{21 \cdot (M_1^2 + M_2^2) - 6 \cdot M_1 \cdot M_2}{4 \cdot W^2 \cdot H^3 \cdot E} \tag{8.1}$$

Where M_1 and M_2 are the applied moments on each of the arms of the DCB specimen, W is the width of the specimen, H is the height of each beam of each arm and E is Young's modulus in the fiber direction. The loading is illustrated in Figure 8.5. In this work the DCB specimens is loaded with equal and opposite bending moments i.e, $M_1 = -M_2$ and $M_2 > 0$, thus loading it in mode I .

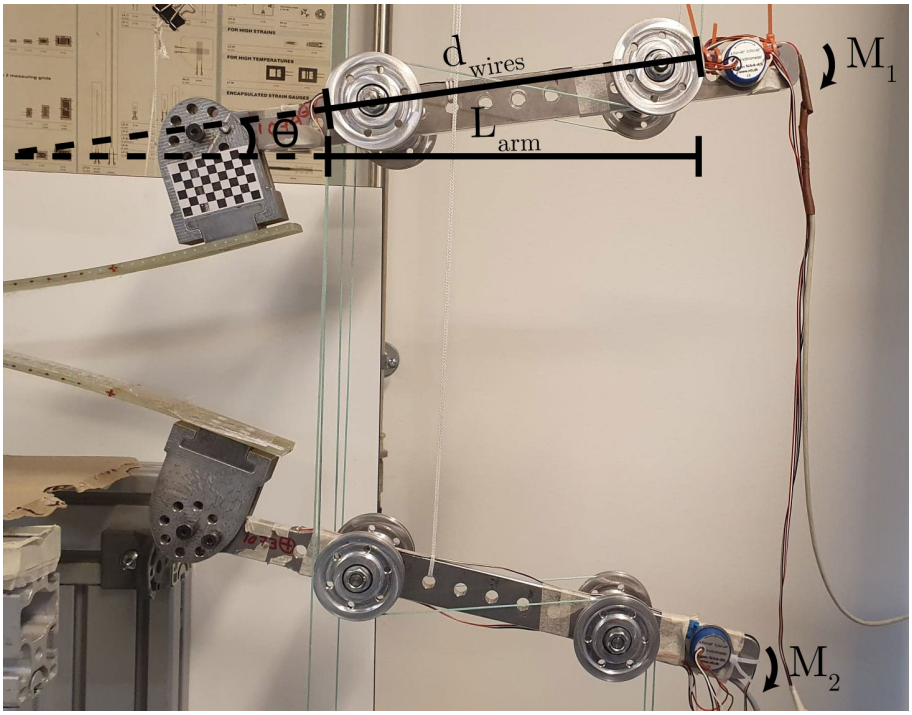


Figure 8.5. *Illustration of loading of the DCB specimen with bending moments and variables used for accounting for rotation.*

The Young's modulus has been determined using three point bending tests on selected DCB specimens with an average value of 30.8 GPa with a maximum deviation of 0.62 GPa. For further details on the derivation of the J-integral see Sørensen et al. (2006).

The applied moments is calculated from Equation 8.2

$$M = \frac{F}{2} \cdot L_{arm} \tag{8.2}$$

Where L_{arm} is the horizontal distance between the wire on the rollers as shown in Figure 8.5, F is the force applied by the tensile test machine.

Due to rotations of the loading arms the horizontal distance between the rollers change during the test, thereby changing the applied moment. This is accounted for by using Equation 8.3.

$$L_{arm} = \cos(\theta) \cdot d_{wires} \tag{8.3}$$

Where θ is the angle measured by the inclinometers, and d_{wires} is the distance between the wires as shown in Figure 8.5.

8.4 Preliminary Experiments

In order to gain experience with the manufacturing and testing of the DCB specimens preliminary experiments are done. The preliminary experiments are further used to check whether the observations made in the parameter study in section 7.3 are valid. Two different types of DCB specimens are designed, manufactured and tested based on the observation from the parameter study.

- **V01:** A virgin type using the general layup. This is made to get a baseline to compare the effect of introducing patches in the DCB specimens with.
- **W01** A weakening type with a weakening patch positioned in the general layup. The patch consists of a PTFE film, that will act in a similar manner as a pre-crack. The patch is positioned between layer 10 and 11, with a longitudinal distance to the pre-crack of 100 mm. The patch has a length of 50 mm.

The DCB specimen with a weakening patch is illustrated in Figure 8.6.

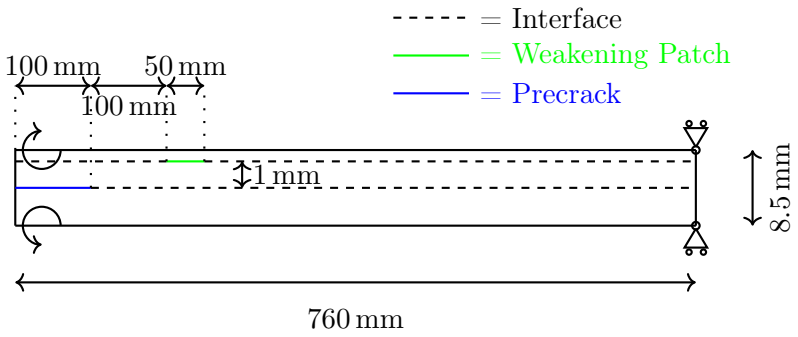


Figure 8.6. *DCB specimen with a weakening patch.*

The preliminary experiments showed that it is possible to onset and propagate multiple delaminations when a weakening patch is used based on visual inspection. However, the range of the clip gauge is too small to measure the crack mouth opening displacement for the entire experiment. The clip gauge simply falls of just as the delamination reaches the patch. Therefore it is not possible to determine whether multiple delaminations has a toughening effect.

One way to alleviate this could be to move the patch closer to the pre-crack. However, it was observed during the experiment of a V01 specimen, that the damage zone with large scale fiber bridging extended approximately 50 mm behind the crack tip. A full damage process zone is needed before the patch is reached, in order to correctly study the effect of patches introduced in the specimens. Therefore, moving the patches closer is not deemed a viable solution.

Therefore, a different method to measure the CMOD is required. This is done in the form of image registration, which will be explained in the next section.

8.5 Virtual Clip Gauge

By using image registration a *Virtual Clip Gauge* can be utilised.

The basic idea of the Virtual Clip Gauge is to measure the CMOD on images captured continuously during the experiments. This is done by marking each arm of the DCB specimen with a cross and using image registration to track the change in position of the crosses throughout the captured images. This is done by tracking the change in position of the crosses between images using the Computer Vision Toolbox from MATLAB. Image registration basically detects the position of a detail selected from one image on a second image. The two images is henceforth referred to as the *target image* and the *search image*, respectively.

The image registration is done using the Computer Vision Toolbox from MATLAB. The image registration in the Computer Vision Toolbox utilises local features. A local feature is a pattern, which is distinct from the surrounding area of the image. Usually this is done using image properties such as texture and intensity (Mikolajczyk and Tuytelaars, 2009). The toolbox contains different feature detectors and descriptors. The feature detectors are used to detect the local features in the image, and the descriptors are used to describe the neighborhood of a local feature. (MathWorks, 2022a)

The process of image registration follows five steps (MathWorks, 2022a):

- Detect local features in both the detail, selected from the target image, and the search image
- Extract descriptors for the local features
- Determine candidate matches by matching the descriptors
- Obtain the location of the candidate matches
- Determine matching points and the geometric transformation

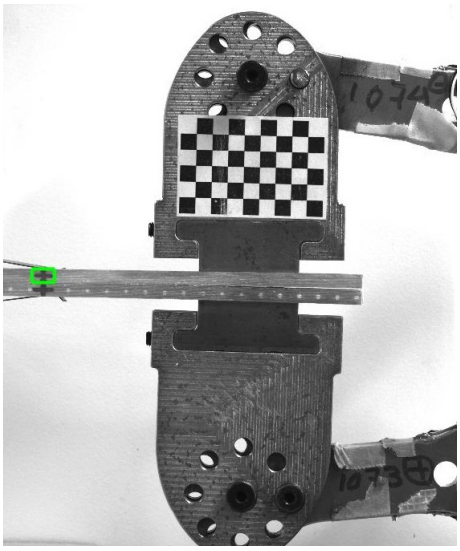
In the following the principle of image registration is explained using the two images shown in Figure 8.7a and 8.7b. The square in Figure 8.7a illustrates the selected detail to find in Figure 8.7b.

First the local features are detected in both the detail, and the search image. The detected local features are illustrated in Figure 8.8. Feature descriptors are then extracted for the local features.

The candidate matches are found by matching the local features from the two images by using their descriptors as shown in Figure 8.9a. The outliers then are removed leaving only matching points as shown in Figure 8.9b. The geometric transformation of the detail is then estimated, to find its location on the search image, which yields a transformation matrix.

The transformation matrix is then used to show the found position and rotation of the detail on the search image as illustrated in Figure 8.10.

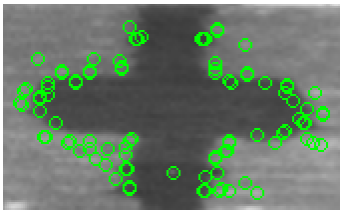
Based on the principles for Image Registration a virtual clip gauge is programmed.Different types of feature detectors is available in the Computer Vision Toolbox. In order to select the best type each of them are tested.



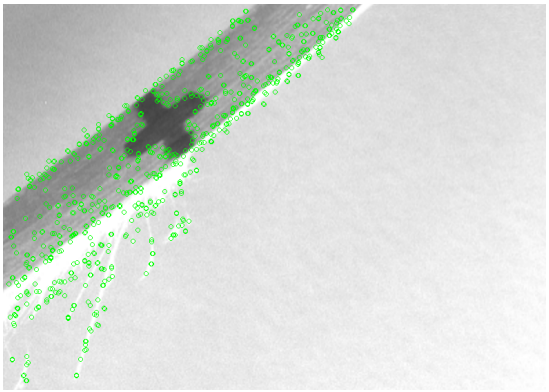
(a) Target image, the green square illustrates the selected detail.

(b) Search image on where to find the detail.

Figure 8.7. *Images used for explaining the principles of image registration.*

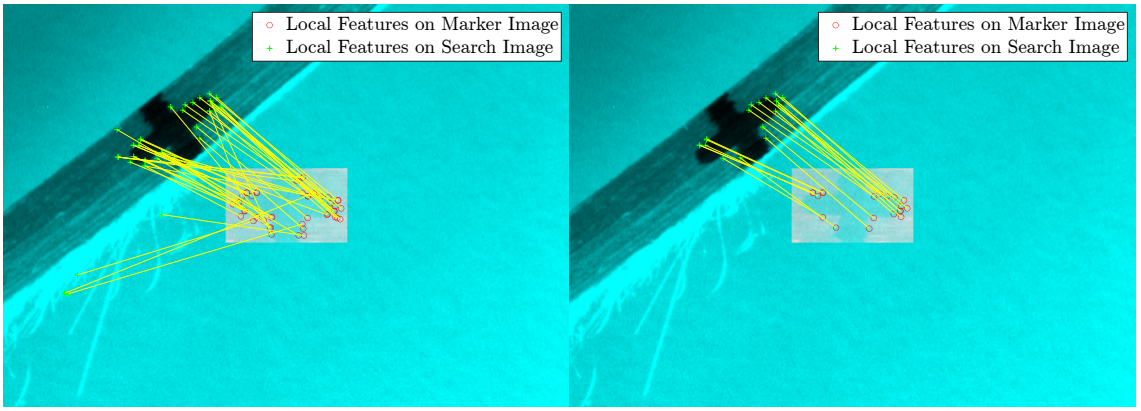


(a) Selected detail.



(b) Search image.

Figure 8.8. *Detected local features on the two images (Circled in green).*



(a) Candidate matches. *(b) Matches after the geometric transformation*

Figure 8.9. *Matching points between the two images.*

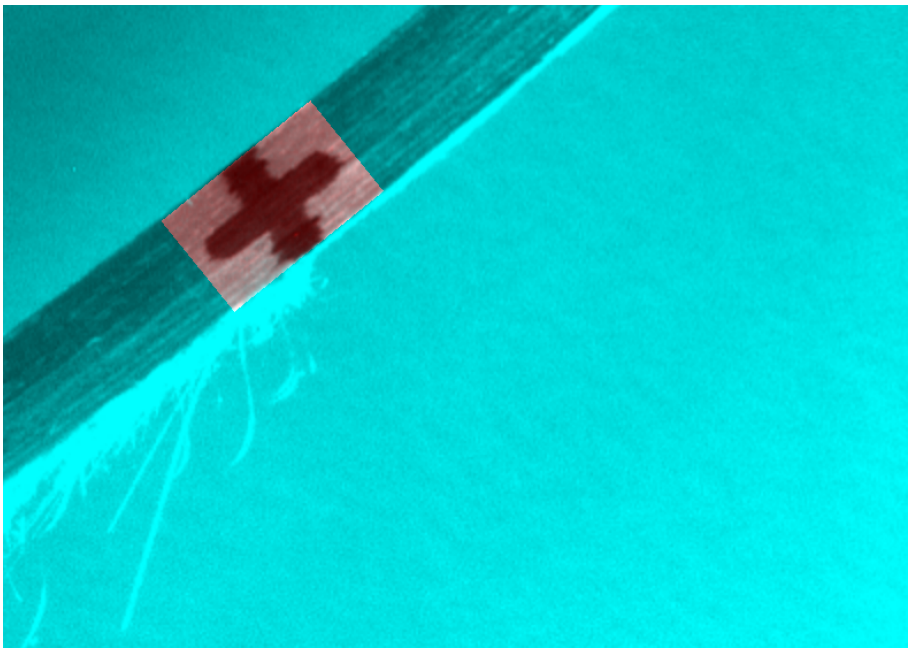


Figure 8.10. *The detail overlaid on the search image.*

From the test three conclusions are made.

- Using one type of feature detectors and descriptors alone will either result in difficulty to find sufficient matches or too high of a computational cost.
- The primary computational cost comes from the extraction of the descriptors. It is therefore beneficial to reuse the extracted descriptors if possible.
- The resolution of the detail has a substantial effect, on how many matching points are found.

Based on aforementioned, four different types of feature detectors and descriptors are used; SURF, BRISK, KAZE and SIFT for further details see MathWorks (2022b). All four are combined into one image registration algorithm shown in Appendix I. The idea of this is to use the faster types first, and then continue with a slower but better algorithm, if not enough matching points are found. Additionally, a region of Interest is used with three of the feature detectors. This is used to further reduce the computational time of the image registration, by reducing the search area from the full picture to smaller part near expected location of the selected detail. Finally the resolution of both the detail and the search image are increased using the MATLAB function *imresize*.

The different types of feature detectors and descriptors are used in the following order:

- SURF with region of interest
- BRISK with region of interest
- KAZE with region of interest
- SIFT
- SURF without region of interest
- BRISK without region of interest
- KAZE without region of interest

The input to the algorithm is two selected details in the form of each cross on the DCB arms, the image on which they are to be found, a region of interest for each detail and

a integer describing the minimum number of matching points required to proceed to the next image.

If a feature detector and descriptor is used without a region of interest, for the first detail, the local features detected on the search image are reused for the second detail. This is possible as the as the search image remains the same for the two details. This reduces the computational cost.

The output from the algorithm is the angle, scale, location and number of matching points found for each detail. These are then used to calculate the center position of each detail and the distance between them. As this distance is in pixels a calibration is needed.

A calibration script has therefore been developed. This is set up to utilise a checkerboard attached to the DCB test rig, in the approximately same plane as the crosses drawn on the DCB specimen. The checkerboard is automatically detected and presented with a numbering for the detected points. An example of the checkerboard with the numbering is illustrated in Figure 8.11.

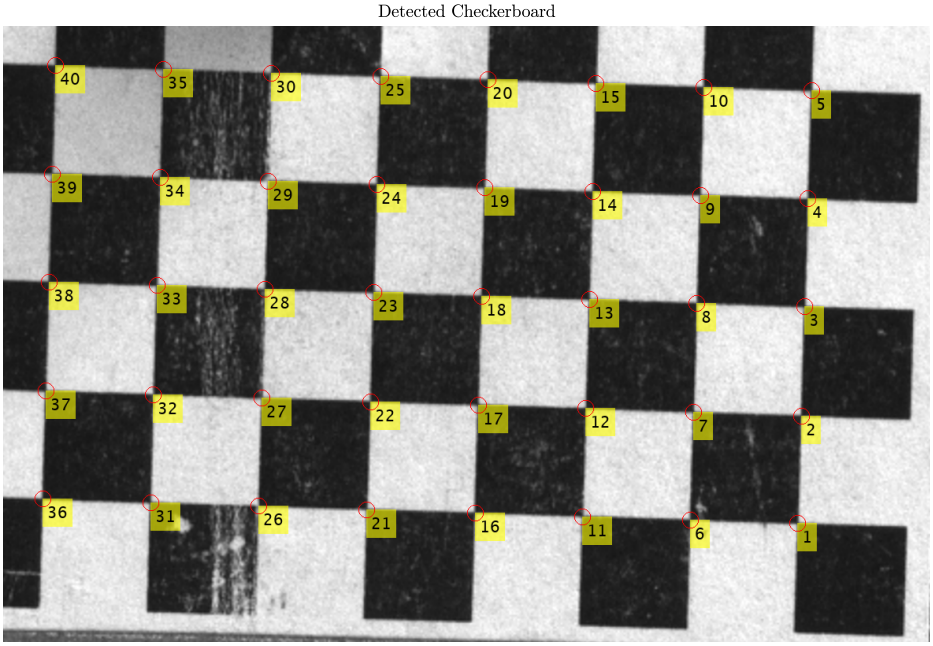


Figure 8.11. *Checkerboard with numbering used for the calibration.*

The two points to measure between are then entered, along the distance between them measured in mm. This is used to calculate a distance per pixel, which used to calculate the CMOD in mm.

The first trials with the image registration algorithm shows, that the light in the images changes throughout the experiment. This caused problems if the light change affects the areas with the two details too much. In order to alleviate this, the two details are updated when the angle found by the image registration algorithm have increased by five degrees for

both details. The updated details are then used until they are updated again. Although the computational cost of updating the details is substantial, it helps in reducing the total computational cost of running the Virtual Clip Gauge on the entire set of recorded images. Furthermore, it makes the Virtual Clip Gauge more robust to change in lighting during the experiments. The script for the Virtual Clip Gauge is shown in Appendix I. An visual example from the Virtual Clip Gauge is illustrated in Figure 8.12. Where the location found for the two details are shown in green, the regions of interest are shown in red for the top detail and blue for the bottom detail.

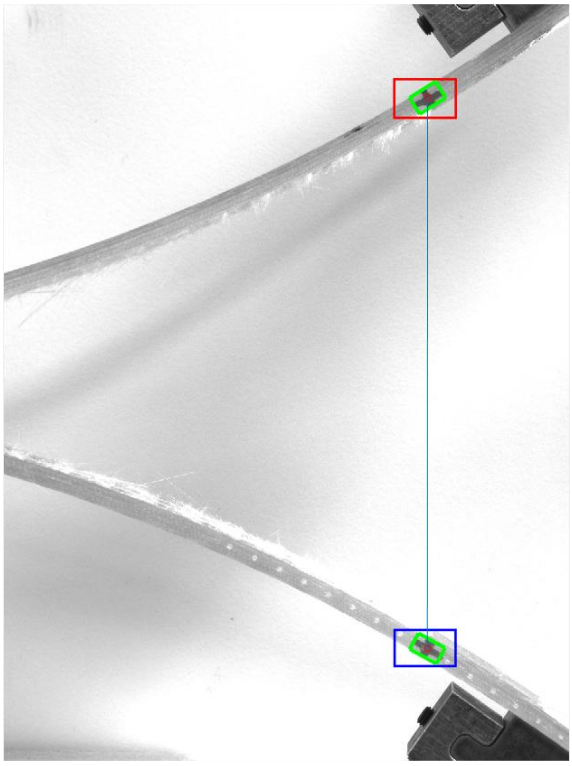


Figure 8.12. *Visual example from the image registration process with the Virtual Clip Gauge.*

A time is needed for each image in order to synchronise the results from the CMOD with the data from the tensile test machine. This is done by taking the timestamp from the first and the last captured image and calculating an average frame rate. This is then used to set a time associated with each CMOD measurement. This is then manually synchronized with the data and time from the physical clip gauge.

In order to confirm that the developed Virtual Clip Gauge works correctly it is compared with the measurements from the physical clip gauge as shown in Figure 8.13.

Based on Figure 8.13 it is concluded that the Virtual Clip Gauge measures the correct CMOD, this is also confirmed by measuring the CMOD with a caliper. The flat part of the CMOD for the physical clip gauge is were it has fallen off.

The Virtual Clip Gauge is used to measure the CMOD in all the remaining test done in this work. Images that are to be used for the Virtual Clip Gauge are captured with a digital camera. Additionally two other cameras are used to record the experiments from two different angles. The final test setup including the additional cameras is illustrated in Figure 8.14

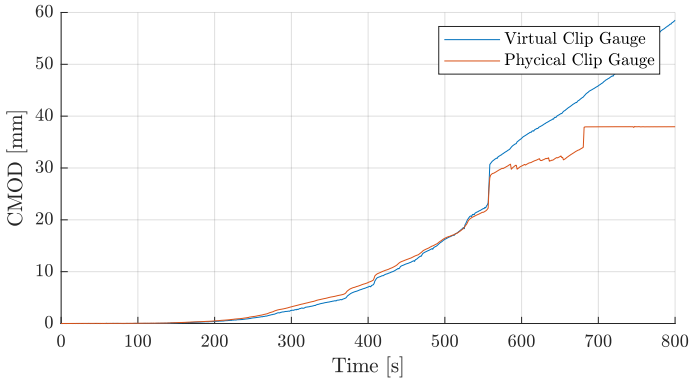


Figure 8.13. *Comparison of Virtual Clip Gauge and physical clip gauge.*

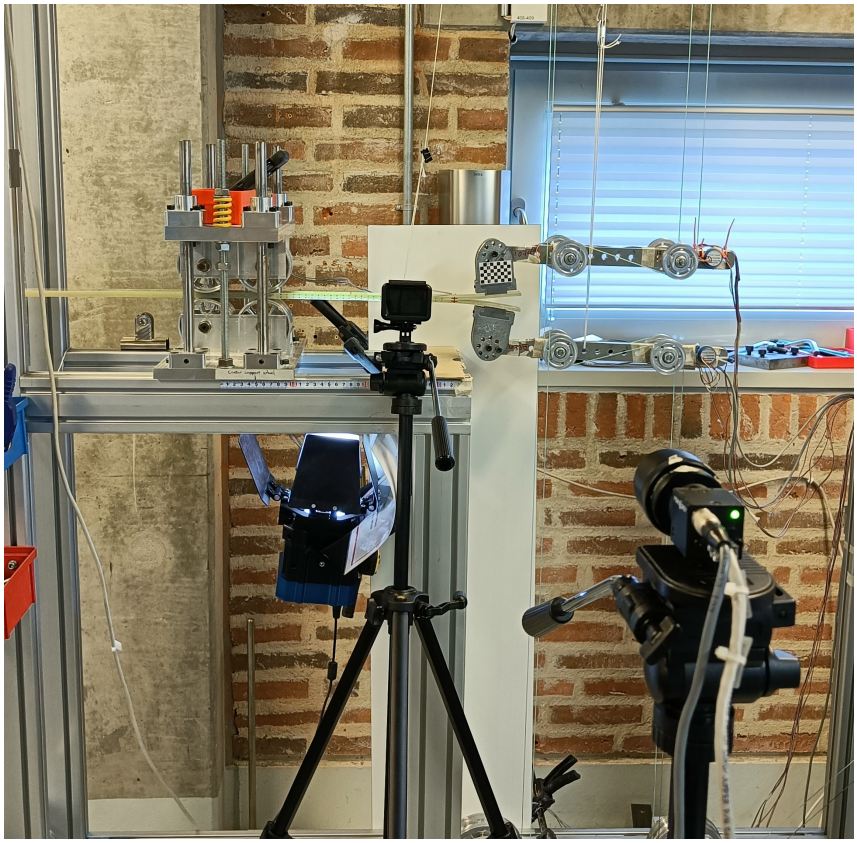


Figure 8.14. *Experimental setup with the test fixture, digital camera for the Virtual Clip Gauge and the two cameras to record the experiments.*

8.6 DCB Specimens Used for The Experiments

In this section the patches and specimens used for the experiments on determining whether multiple delaminations is promoted are presented.

In addition to the two types of DCB specimens used in the preliminary experiments presented in section 8.4, four new types are manufactured. Three with toughening patches and one virgin type. The three toughening patches are explained in the following:

Fibertex FICHE 45 The patch is a nonwoven fiber mat from Fibertex called *FICHE 45*.

The composition of the mat is 60 % polyester and 40 % polypropylene, which have been needle punched to bond the fibers. Additionally the mat have been melted on one side. The patch is an interleave type.

Fibertex VIGONIT 30 The patch is a nonwoven fiber mat from Fibertex called *VIGONIT 30*. The composition of the mat is 100 % polyester, which has been calendered to bond the fibers. Additionally the mat has been melted on one side. The mat is positioned in the main interface as a toughening patch. The patch is an interleave type.

Stitching with aramid thread This patch is made by stitching the two center layers together. The stitching operation is done using an aramid thread, with the lines of stitching transverse to the fiber direction in the mats. The stitching lines have a spacing of 4 mm, and the entire patch has a length of 52 mm. The patch is a through-the-thickness type.

The toughening patches is inserted into the general layup described in section 8.1 in the main interface as illustrated in Figure 8.15.

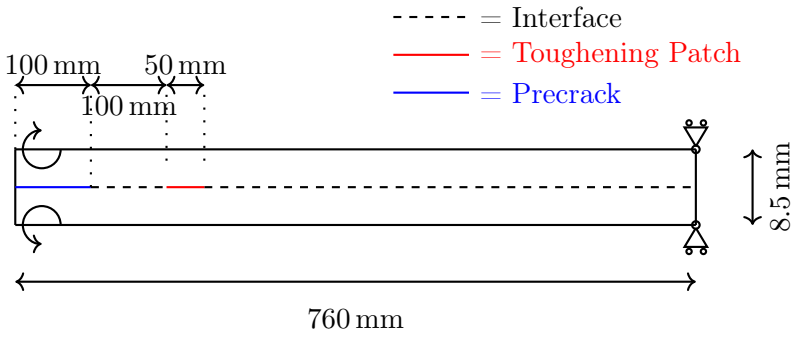


Figure 8.15. *DCB specimen with placement of the toughening patch. The illustration is not to scale.*

The four new types of DCB specimens are manufactured in two composite plates. e.g. by manufacturing one plate with one half being with the T01 DCB specimens and the other half being the T02 DCB specimens. The last two types DCB specimens are manufactured in the same manner. The DCB specimens are manufactured as explained in section 8.1.

Thereby, the different types of DCB specimens, including the two from the preliminary experiments, which will be used for determining whether multiple delaminations are promoted, are as follows:

- V01: A type without any patches. The pre-crack has a length of 75 mm.
- V02: A type without any patches. The pre-crack has a length of 100 mm.
- W01: A type with a weakening patch consisting of PTFE film. The patch is positioned between layer 10 and 11, with a longitudinal distance to the pre-crack of 100 mm. The patch has a length of 50 mm. The pre-crack has a length of 100 mm.
- T01: A type with a toughening patch consisting of a FICHE 45 mat from Fibertex. The patch is positioned in the main interface, with a longitudinal distance to the pre-crack of 100 mm. The patch has a length of 50 mm. The pre-crack has a length of 100 mm.
- T02: A type with a toughening patch consisting of a VIGONIT 30 mat from Fibertex. The patch is positioned in the main interface, with a longitudinal distance to the

pre-crack of 100 mm. The patch has a length of 50 mm. The pre-crack has a length of 100 mm.

- T03: A type with a toughening patch is manufactured, by stitching the two center layers together with an aramid thread. The patch is positioned in the main interface, with a longitudinal distance to the pre-crack of 50 mm. The patch has a length of 50 mm. The pre-crack has a length of 100 mm.

9 Experimental Results

In the following, results from the experiment are presented. As the main focus in this work is to increase the toughness by promoting multiple delamination, the fracture resistance is plotted as a function of the crack mouth opening displacement. Furthermore fractography is used to study the specimens.

Based on the observations in section 7.3, it is expected, that the fracture resistance curves can be divided into distinct phases. There are five distinct phases for the specimens with patches, whereas the virgin specimens are expected to have the first three phases. The phases are listed below and illustrated in Figure 9.1:

- Phase 1: Fracture resistance increases without any noticeable delamination crack growth.
- Phase 2: Delamination crack growth initiates and is followed by an increase in the fracture resistance caused by the development of a damage zone with fiber bridging. In this phase the damage zone continuously increases in size leading to an increased toughness.
- Phase 3: The fracture resistance approaches a steady state value. In this phase a full damage zone is developed and delamination crack growth occurs in a almost self-similar manner.
- Phase 4: Fracture resistance and delamination crack growth becomes affected by the patch, as the crack tip advances close to or into the patch.
- Phase 5: Fracture resistance approaches a steady state value, after the crack tip has advanced beyond the patch. Comparing the fracture resistance in this phase with the fracture resistance in phase 3 will reveal if multiple delamination has occurred.

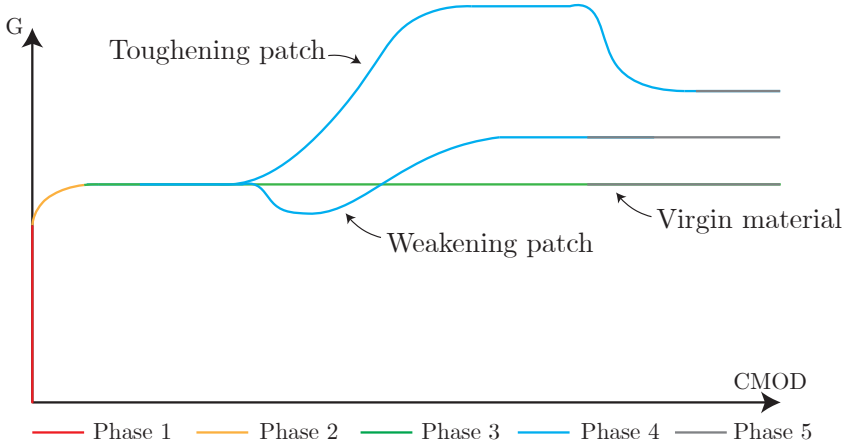


Figure 9.1. *Fracture resistance curve with the five expected phases.*

The results from the experiments of one DCB specimen from each type are illustrated in Figure 9.2 and 9.3. In Figure 9.2, the results from the entire experiments are shown, whereas Figure 9.3 is a zoom of the first three aforementioned phases. In the following, the virgin experimental results are explained first and afterward the weakening patch and each of the toughening patches are explained individually.

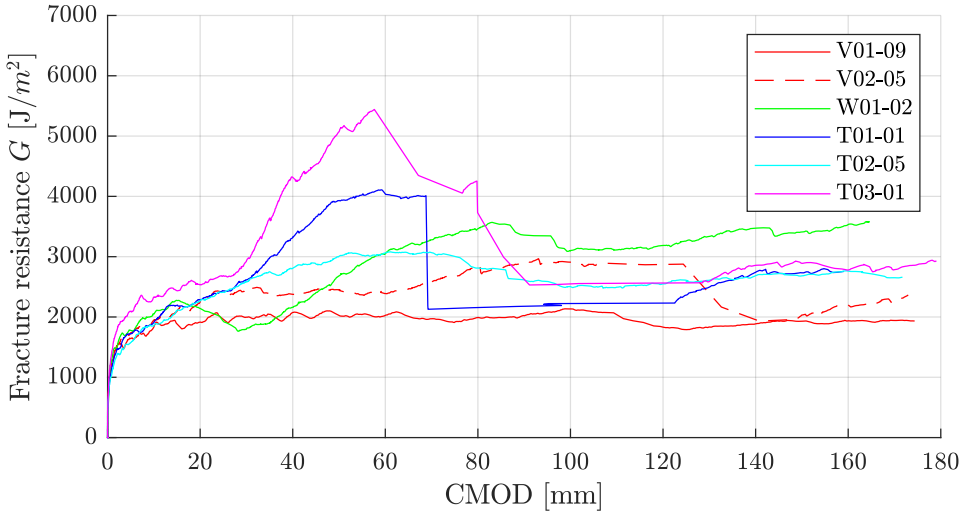


Figure 9.2. *Fracture resistance curves for the five types specimens.*

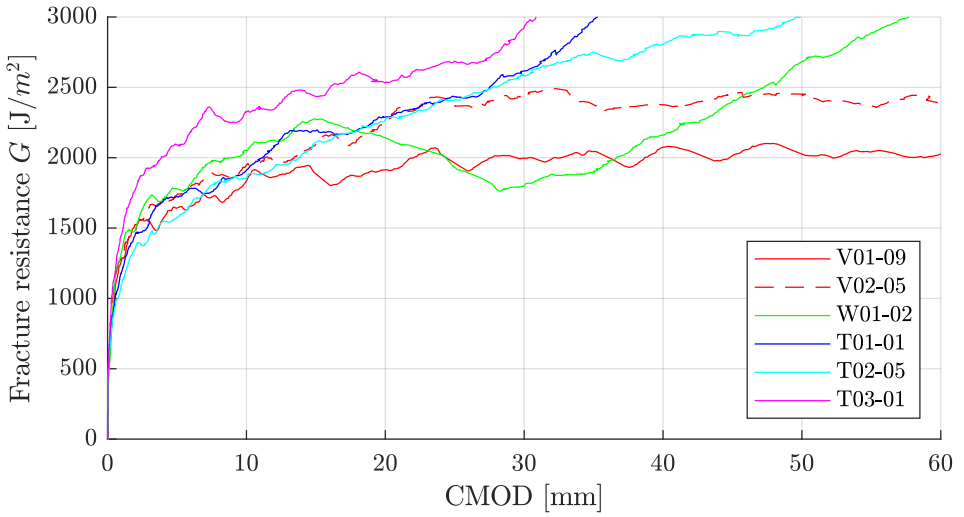


Figure 9.3. *Fracture resistance curves for the five types specimens in the first three phases.*

The virgin specimens are used for the comparison of the effects of using different patches. This is mainly for phase 3 and 5, as it can be difficult to assess, whether a steady state value for the fracture resistance in these phases are reached for some of the DCB specimens. Thus, if multiple delamintions has been successfully onset this is expected to be seen in the fracture resistance in phase 5.

V01 and V02: Results from the V01 specimens are illustrated in Figure 9.4. As mentioned above, the first three phases are expected to be seen in the plot, which is also the case.

Especially, the two V01-specimens exhibits close to stable crack growth in phase 3. The fracture resistance in phase 3 is approximately 2000 J/m^2.

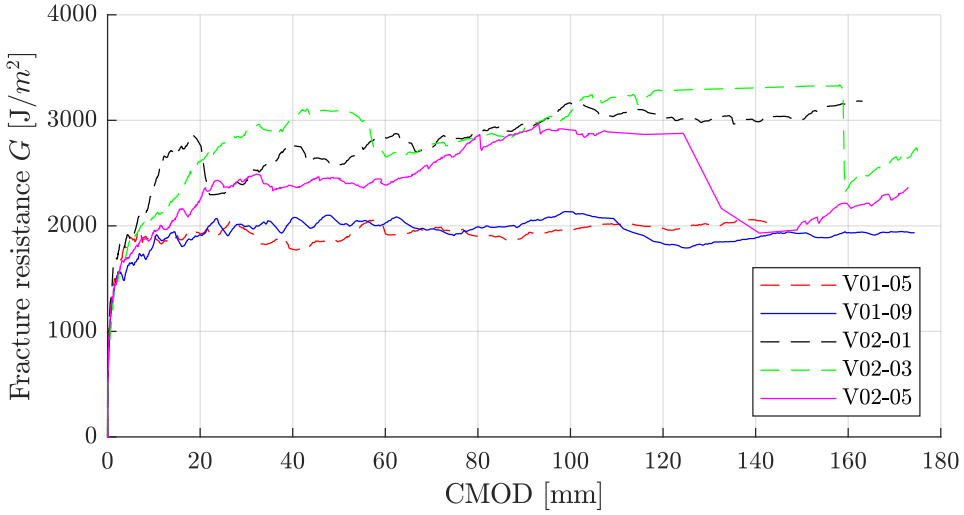


Figure 9.4. *Fracture resistance curves for the V01 and V02 specimens.*

On the other hand, the V02 specimens have larger variations in the fracture resistance and also seem to exhibit unstable crack growth, identified by the abrupt drops in the fracture resistance. It is therefore difficult to assess a steady state fracture resistance common for all three V02 specimens. The V02-05 seem to be approaching a steady state fracture resistance in phase 3 is approximately 2400 J/m^2 in the range of CMOD 20 mm-60 mm. However, it is difficult to assess whether it the correct steady state value, as the fracture resistance increases again afterwards.

The large variations in the fracture resistance for the three V02 DCB specimens may be explained by the fact, that they were manufactured in the same plate as the ones with the T03 DCB specimens. The stitching operations done on the T03 half of the plate may have distorted or damaged the fibers in the V02 specimens, which ultimately leads to the larger variation in the fracture resistance.

There is a difference in the steady state fracture from the V01 specimens of 2000 J/m^2, to the value of 2400 J/m^2 for the V02-05 specimen. This is explained by the two composite plates being manufactured at different times. The V01 specimens was cured longer, than the other specimens. Thereby, making the epoxy more brittle leading to the lower fracture resistance than in the V02 specimens. As it is unknown if the fracture resistance 2400 J/m^2 is a correct steady state value for the V02 specimens, the steady state value for the V01 will be used a comparison point, when evaluating if multiple delaminations has successfully been onset in the other types of specimens. However, keeping in mind that the fracture resistance of 2000 J/m^2 might be lower, because of the longer curing.

W01: The results from the DCB specimens with a weakening patch are illustrated in Figure 9.5. Phase 1 and 2 are easily identified from the figure, whereas the steady-state behaviour in phase 3 is less obvious, as each of the W01 specimens reaches a maximum value and drops off again shortly after. Therefore the value of 2000 J/m^2 from the V01 specimens are used.

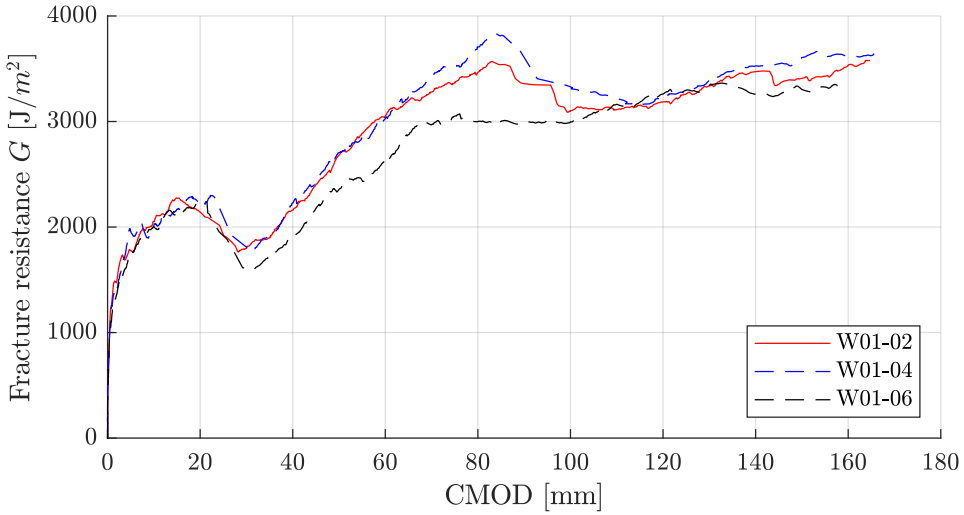


Figure 9.5. *Fracture resistance curves for the W01 specimens.*

From Figure 9.5 it is evident that phase 3 is followed by a drop in the fracture resistance. This is consistent with the expected behavior for a weakening patch as shown in Figure 9.1, and is explained by a decrease in the load bearing capacity, when entering the area with a weakening patch. Afterwards the fracture resistance increases in a steady manner, until the fracture resistance reaches an average peak value of approximately 3400 J/m^2 for the three specimens. This increase is caused by the crack tip in the secondary interface, which starts to advance forward. Thus, energy is spend on advancing both crack tips, and this has a toughening effect.

In phase 5 the steady state fracture resistance is approximately 3200 J/m^2. Comparing this with the steady state fracture resistance from the V01 specimens indicates that both crack tips are still active and dissipate energy. Taking into account, that the V01 specimens was cured longer, is it estimated that the fracture resistance for the W01 have increase by approximately 50 %, by promoting multiple delaminations.

To inspect that multiple delaminations is actually occurring, the eperiments was visually inspected. A picture of the multiple delamination is shown in Figure 9.6.

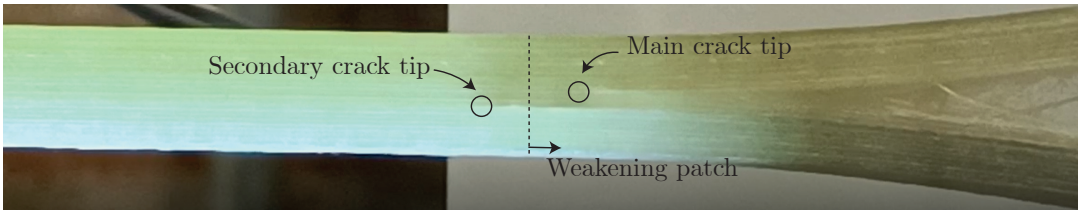


Figure 9.6. *Figure of W01 specimen viewed from the side and shows two crack tips.*

From Figure 9.6 it is evident that multiple delaminations indeed are onset and starts propagating.

The drop in fracture resistance at a CMOD of 84 mm for W01-02 and W01-04 is explained by a sudden increase in crack growth in the main crack. It is observed that the secondary crack advances out of the patch with the main crack lagging behind. This occurs in a stable

manner. However, once the secondary crack has advanced out of the weakening patch, and the main crack is still in the patched area, the main crack jumps, which places it in front of the secondary crack. The jump in the main crack is believed to cause the drop in the fracture resistance at 84 mm in Figure 9.5. Aforementioned behaviour was not observed in the W01-06 specimen, which explains its more steady value in phase 5. Common for all of them are, that the secondary crack seems to be lagging behind the main crack, once the main crack has advanced out of the patch.

It is also observed in the experiments, that the rate at which the secondary crack grows decreases and almost comes to a stop, once the main crack overtakes the secondary crack.

Additionally, the DCB specimens have been investigated by fractography after the experiments. In Figure 9.7 a fractographic image taken at the end of the weakening patch, is shown. The red line that is seen in this image is the PTFE film, used to create the weakening patch. The line separating the light area and the darker area is the secondary crack. As shown in the image, the secondary crack advances out of the weakening patch and thereby creating multiple delamination.

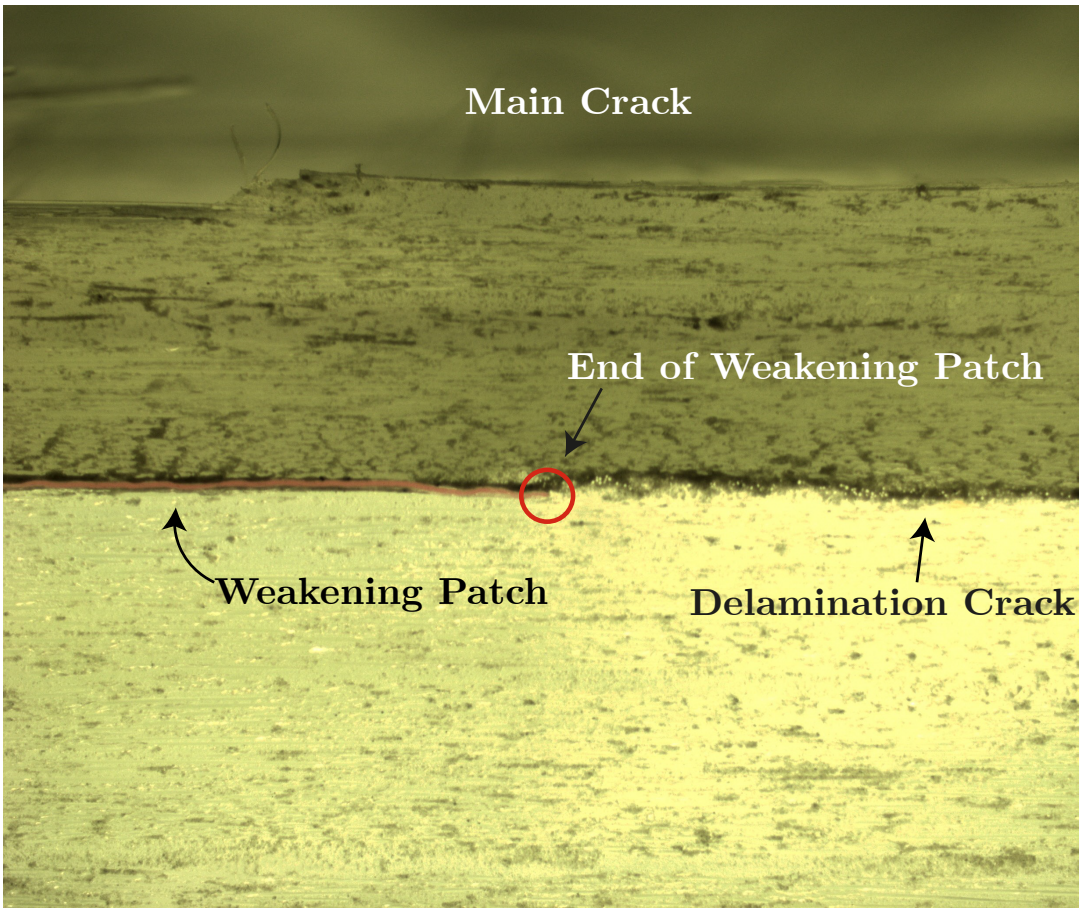


Figure 9.7. *Fractographic inspection of the weakening patch in a W01 specimen. End of weakening patch marked on the specimen.*

In Figure 9.8 an fractographic image of the crack tip in the secondary interface is shown. Similar to the observations for Figure 9.7, a line separating the light and the dark area is observed. Furthermore, where the line stops indicates where the crack tip is in the specimen. This will be used as a comparison point, when examining other specimens for

multiple delaminations.

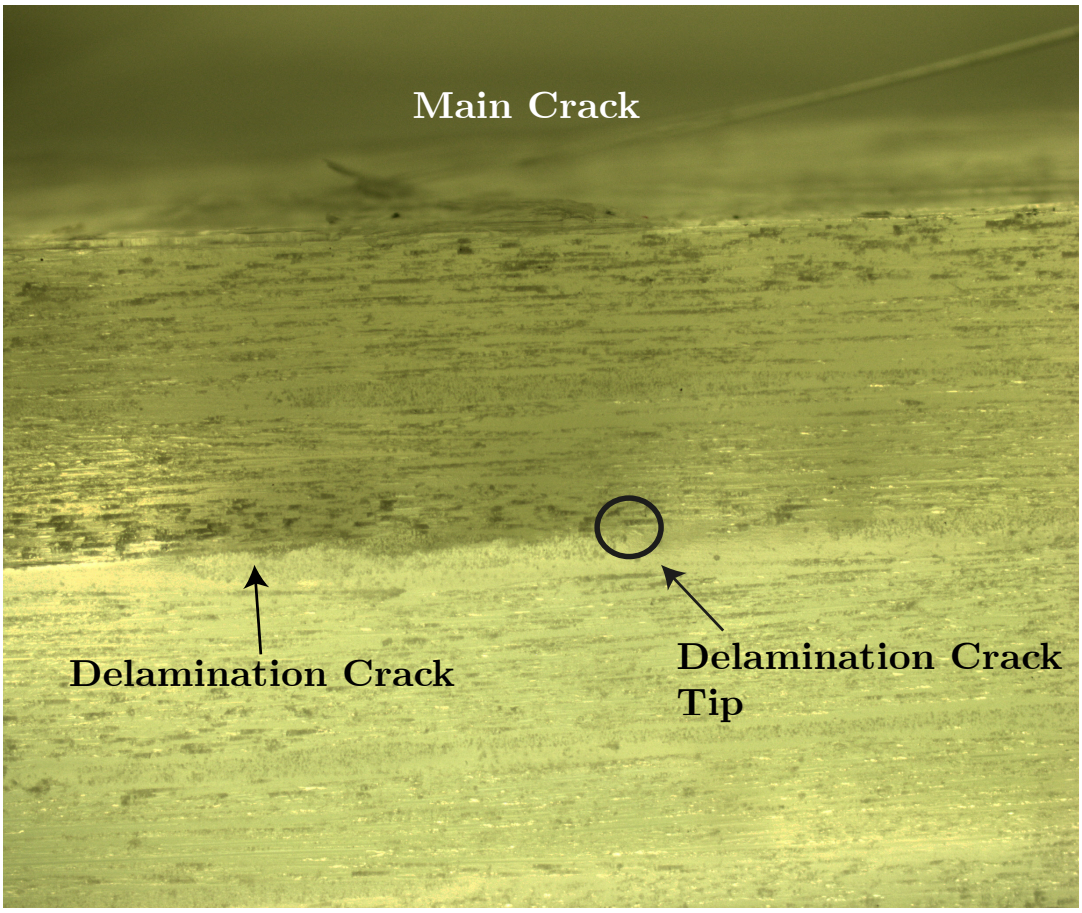


Figure 9.8. *Fractographic inspection of the secondary crack in a W01 specimen. Crack tip in the secondary interface is marked on the specimen.*

T01:

The results from the DCB specimens with a T01 toughening are shown in Figure 9.9. Again the first two phases are easily identified, whereas the steady-state behaviour in phase 3 seems to be missing.

Looking at Figure 9.3 it is clear that the T01-01 specimen does not reach a steady-state fracture resistance but keeps increasing in phase 3. However, the increase is only gradual for a CMOD value of 5 mm-30 mm and almost comparable to the other DCB specimens.

As the T01 specimens does not reach a steady state fracture resistance before entering the patch, the value of 2000 J/m^2 from the V01 specimens are used. However it is expected that it should be higher in the T01 specimens as they was not cured as long as the V01 specimens.

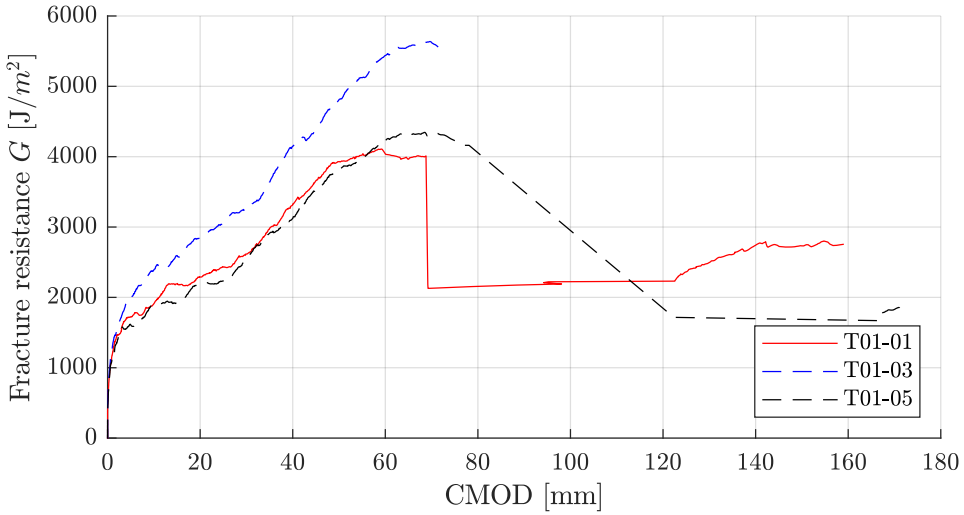


Figure 9.9. *Fracture resistance curves for the T01 specimens.*

In phase 4 a large increase in fracture resistance occurs due to the main crack entering the toughening patch. Thus, the toughening patch is beneficial for toughening the interface. The T01-01 and T01-05 specimens reaches a peak fracture resistance of 4200 J/m^2 and 4200 J/m^2, with the T01-03 specimen reaching a peak value of 5600 J/m^2. However, once the crack tip advances out of the patch, the toughening effect decreases and rapid crack growth occurs. This was observed during the experiment on T01-01, where the crack tip moved a distance of approximately 50 mm in one step.

The rapid crack growth, is followed by a similar increase in the CMOD. During this crack growth, only a few images are captured by the camera. The Virtual Clip Gauge can therefore not provide reliable CMOD-values, until the crack grows in a more stable manner. The unstable crack growth caused the crosses, used for the Virtual Clip Gauge, on specimen T01-03, to be placed outside the view range of the camera. The same was encountered for the T01-05 specimen, where only a few data points was recorded after the unstable growth. Thus, only the T01-01 specimen is considered reliable after the rapid growth. However, the data points in the interval from 70 mm-123 mm in Figure 9.9 cannot be used.

Considering the T01-01 specimen in phase 5, when stable crack growth is restored, the fracture resistance has value of approximately 2700 J/m^2.

This is an increase compared to the steady-state fracture resistance in phase 3 for the V01 specimens. However, as mentioned previously, the fracture resistance in phase 3 for the T01-01 is expected to be higher, based on the shorter curing time compared to the V01 specimens. This could therefore indicate that is has not been possible to onset new delaminations.

This was confirmed by visual inspection during the experiments and by the recorded videos afterward. In order to finally confirm this, the T01-01 specimen was further investigated by fractography. The fractographic images from the inspection are shown in Figure 9.10. Each image is captured at different locations along the length of the specimen, one aligned with the center of the patch, one aligned with the transition from the patch to the virgin

material and one 40 mm ahead of the toughening patch. It is difficult to observe the toughening patch, the images are therefore captured based on the measurements from the manufacturing of the specimens and the untested T01-04 specimen.

No new delaminations was observed on the images. Thereby confirming that it has not been possible to on new delamination cracks in the T01 specimen.

T02:

The fracture resistance curves for the T02 specimens are shown in Figure 9.11. The tendencies for phase 1 and 2 are similar to the T01 specimens. However with some fluctuation being present in the T02-01 and T02-03. All three T02 specimens seem to go from phase 2 directly to phase 4. The steady state fracture resistance of 2000 J/m^2 from the V01 specimens, is therefore used as an approximate steady state value for the T02. Again it is expected to be higher for the T02 specimens, as the curing process was shorter, than for the V01 specimens. The tendencies in phase 4 are similar to the other specimens with toughening patches. The average peak fracture resistance is approximately 3200 J/m^2.

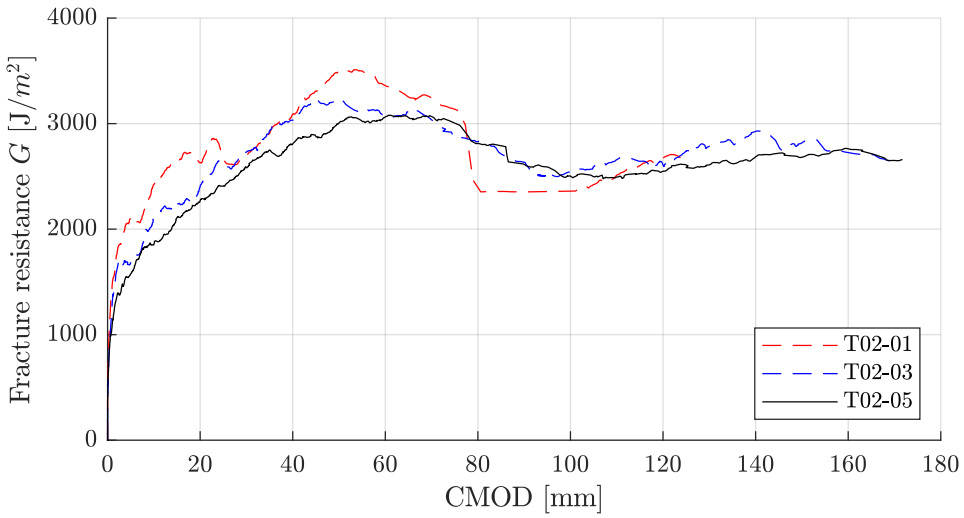
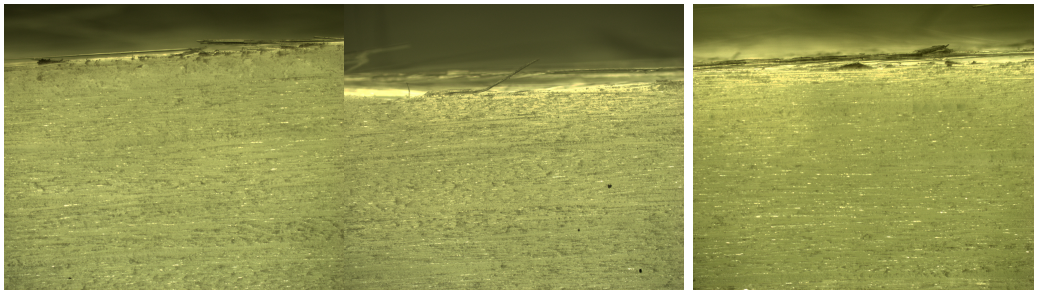


Figure 9.11. *Fracture resistance curves for the T02 specimens.*

The transition from phase 4 to phase 5 is more steady compared to the T01 specimens,



(a) *Center of the toughening patch.* ***(b)*** *Front of the toughening patch.* ***(c)*** 40 mm *ahead of the toughening patch.*

Figure 9.10. *Fractographic inspection of T01-01 specimen.*

with the T02-01 behaving differently. The more steady transition is assumed to be due to peak fracture resistance being closer to the steady-state facture resistance in phase 5.

In phase 5 the T02-03 and T02-05 specimens approaches a steady-state fracture resistance, whereas T02-01 was stopped before it reached a steady-state fracture resistance. The steady state value in phase 5 is approximately 2700 J/m^2. This is an increase compared to the steady-state fracture resistance from the V01 specimens. However, as mentioned previously, the fracture resistance in phase 3 for the T02 specimens is expected to be higher, based on the shorter curing time compared to the V01 specimens. This indicates, that it has not been possible to onset new delamination cracks, which is confirmed by the observations during the experiments, and by fractographic inspection on the tested specimens.

T03:

The results from the T03 specimens are shown in Figure 9.12 and the same tendencies as in the other toughening patches are observed. Equivalent to the other toughening patches a steady-state fracture resistance is reached in phase 3. Again the steady-state fracture resistance from the V01 specimens are used as an approximate fracture resistance value in phase 3 for the T03 specimens.

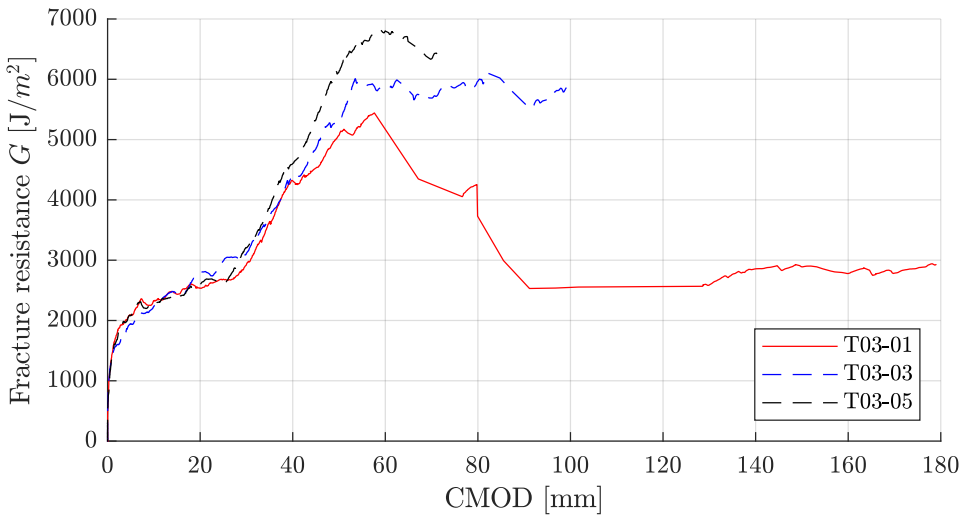


Figure 9.12. *Fracture resistance curves for the T03 specimens.*

When the crack enters the toughening patch in phase four the fracture resistance increases to an approximate average peak value of 6000 J/m^2. Based on the results from all the toughening patches, stitching is the most effective toughening patch used in this work, as it has the highest fracture resistance measured.

As experienced for the T01 toughening patch, the peak on the curves in Figure 9.12 is followed by unstable crack growth. This is observed during the experiment of the T03-01 specimen, where the crack tip advance 70 mm in on step.

The experiments with the T03-03 and T03-05 specimens where stopped, as the crosses, used for the Virtual Clip Gauge, was out of the range for the camera

Due to the aforementioned reasons only the T03-01 experiment is considered, when

evaluating the fracture resistance in phase 5, where it attains a value of $2800\,\mathrm{J/m^2}$. This is higher than the steady-state fracture resistance from the V01 specimens. However, the fracture resistance in phase 3 for the T03 specimens is expected to be higher, as the V01 specimens was cured for a longer time than the T03 specimens. This indicates, that it has not been possible to onset new delamination cracks, which is confirmed by the observations during the experiments, and by fractographic inspection on the tested specimens.

To summarize the observations during the experiment and the results show the following:

- A weakening patch consisting of a PTFE film promotes multiple delaminations, and the multiple delaminations increases the fracture resistance by approximately 50 % in the specimens.
- The toughening patches were not able to promote multiple delaminations in the DCB specimens.
- The toughening patches toughened the specimens, thereby increasing the fracture resistance. However, the toughening were only present, when the delamination in the main interface was within the patch.
- A rapid crack growth was observed for the DCB specimens with toughening patches. This occurred as the delamination advanced close to the end of the patch. The magnitude of this rapid rack growth depended on the magnitude of toughening introduced by the patch.

Discussion 10

In the following the results and methods used in this thesis are discussed.

Arc-length solver: The four arc-length solvers implemented in this thesis are partly done due to a learning goal set forth by the authors. However, it is also needed to model a moment loaded DCB specimen in the MATLAB Framework. Additionally, it generalises the finite element tool developed by Trabal et al. (2022b), which is beneficial if it is to be used in future projects.

Two energy based arc-length solvers are investigated and implemented. However, only one of the energy dissipation based arc-length solvers is strictly needed for the purpose of modelling delaminations in this work. Nevertheless, time and effort has been spend on implementing and debugging both arc-length solvers. Especially, time has been spend on the local energy solver due to the robustness issues mentioned in chapter 5, even though the global arc-length solver is sufficient for modelling delamination problems. This is done, as it is a novel solver, which generalises the problem of modelling energy dissipating mechanisms contrary to the global arc-length solver, which is problem dependant.

The next step in the implementation of the local energy arc-length solver is to locate the error causing the robustness issues. Based on the authors experience, it is to be found in the analytical expression for the constraint derivative, but the exact error is not located. Additionally, it is found that calculating the constraint derivatives with a central difference method solves the problems encountered. Thus, the implementation of the local energy arc-length solver could be based on central differences. This requires that the perturbation size is further investigated. The reason for using analytical expressions in the first place was, that using a central difference approach was expected to increase the computational time. However, this does not seem to be the case for the models done in this project.

Ideally, the implemented arc-length solver should have been used to simulate a forced controlled, moment loaded DCB specimens as this is close to the actual boundary conditions in the test setup. However, this has not been done due to time constraints.

Promoting multiple delaminations by the use of patches

The experiments done with the W01 DCB specimens shows that a weakening patch, consisting of a PTFE-film, promotes multiple delaminations. There was a drop in load bearing capacity, when the delamination in the main interface reached the weakening patch. However, this was followed by an increase in fracture resistance, caused by the secondary delamination starting to propagate. The use of a PTFE-film as a weakening patch, is therefore observed to increase the fracture resistance of the DCB specimen.

A PTFE-film acts similar to setting τ^0 and G_c equal to zero, this is an extreme case of a

weakening patch. However, it was used as a way to increase the probability that multiple delamination would occur. It would have been beneficial to investigate more moderate weakening patches, to check what characteristics would be required of a weakening patch in order to still promote multiple delaminations. However, this was not done in this work as other weakening methods was not found. Furthermore, it was assessed that weakening patches are less likely to be used, as it is generally not favorable to weakening a structural component.

The toughening patches used in this work did not promote multiple delaminations in the DCB specimens. The patches toughened the DCB specimens, when the delaminations propagated inside the patch. However, this toughening effect was lost, when the delamination propagated out of the patch. Furthermore, when the delamination propagated out of the patch, a rapid unstable crack growth occurred followed by a similar increase in CMOD. A drop in load bearing capacity, of the magnitude observed for some of the specimens with a toughening patch, would be unacceptable in a structure, especially if multiple delaminations are not promoted.

A steady-state fracture resistance was not observed in any of the toughening patches used in the experiments. This is caused by the toughening patches being to short, thereby not allowing a full damage zone to develop. It is assessed that using a longer toughening patch would have increased the probability of promoting multiple delaminations. This was not done in this work due to time constraint. Additionally, using a virgin material with a lower toughness, might also increase the probability of promoting multiple delaminations.

Experiments in test setup:

All experimental tests, in this work is done by applying equal and opposite moments to the DCB specimen, as explained in section 8.3. The advantage of using this test method is, that the crack tip does not need to be tracked throughout the test, in order to determine the fracture resistance.

However, the effect of promoting multiple delamination can also be evaluated based on load-displacement curves. Thus, the tests could have been done directly in the tensile test machine as wedge loaded DCB specimens. In this case an increased fracture toughness will yield an increased load bearing capacity.

Further advantages of using the tensile test machine is, that it allows for modelling the exact same boundary conditions, as in the MATLAB Framework. However, an arc-length solver is still needed in order to trace the entire non-linear load-displacement curve.

Additionally, it is beneficial to calculate the fracture resistance in order to characterise the increase in fracture resistance, which can be compared with other toughening methods presented in the literature.

The fracture resistance data gathered from the experiments can further be used to determine onset traction and critical energy release rates to be used in a traction separation law. This was one of the main arguments for using the test method of applying equal and opposite moments to the DCB specimen in this work. The intention was to extract above mentioned values and then use them together with the developed arc-length solver in a parameter study. Due to time constraints this has not been possible in this work, but could be done in a future project.

Conclusion 11

This master thesis is based on the problem formulation stated below:

> *"How can multiple delaminations in fiber composites be promoted by the use of toughening/weakening patches in order to improve damage tolerance in quasi-static loading?"*

In order to answer the problem statement, this thesis is divided into two overall parts. Part one centers around a numerical implementation of advanced solutions techniques in a finite element framework. Part two tries experimentally to verify that the damage tolerance can be improved by onsetting multiple delaminations cracks for DCB-specimens loaded in mode I. The numerical part of this thesis is mainly included as a learning goal.

Four different arc-length solvers are implemented in a finite element framework written in MATLAB. First an arc-length solver based on Crisfields method is implemented and validated. This method encounters convergence issues when a high degree of material instability occurs. Additionally, an arc-length solver based on Riks' method is implemented and validated. Next, two arc-length solvers, which are based on the energy dissipation in the model are implemented. The first solver evaluates the energy being dissipated at element level, but has robustness issues, when using too large or too small step sizes. The cause of the robustness issues has been found to be related to the derivative constraints. However, the exact problem has not been located. The second solver evaluates the dissipated energy being dissipated at a global level and works satisfactory.

Preliminary studies are done in the MATLAB based finite element framework to determine a patch placement for the DCB-specimens, which ensures onsetting of multiple delamination. To do this a mesh dependency study is first done to ensure correct modelling of the delamination problem. This is followed by analyses, which conclude that multiple delamination can be onset by either introducing a weakening patch with interface properties equal to 0 or a toughening patch with interface properties twice the value of the virgin material. Finally, the patch length and the interface wherein the patch is placed is investigated and used for the DCB specimens.

Finally, DCB specimens with weakening patches and toughening patches are manufactured and tested. In order to do this a virtual clip gauge is developed to track the crack mouth opening displacement during the test. Based on the experiments it can be concluded that it is possible to onset multiple delaminations, by having a weakening patch in the layup. The steady state fracture resistance is improved is increased by 50 %, thus improving the damage tolerance. It was not possible to onset multiple delamination cracks in the

DCB-specimens with toughening patches.

Bibliography

Alfano and Crisfield, **2001**. G. Alfano and M. A. Crisfield. *Finite element interface models for the delamination analysis of laminated composites: mechanical and computational issues.* International Journal for Numerical Methods in Engineering, 50 (7), 1701–1736, 2001. doi: https://doi.org/10.1002/nme.93.

Bak, **2015**. Brian Lau Verndal Bak. *Progressive Damage Simulation of Laminates in Wind Turbine Blades under Quasistatic and Cyclic Loading*, 2015. ISSN 22461248.

Boller and Buderath, **2007**. Christian Boller and Matthias Buderath. *Fatigue in aerostructures—where structural health monitoring can contribute to a complex subject.* Philosophical transactions of the Royal Society of London. Series A: Mathematical, physical, and engineering sciences, 365(1851), 561–587, 2007. ISSN 1364-503X.

Braga et al., **2014**. Daniel F.O. Braga, S.M.O. Tavares, Lucas F.M. da Silva, P.M.G.P. Moreira and Paulo M.S.T. de Castro. *Advanced design for lightweight structures: Review and prospects.* Progress in Aerospace Sciences, 69, 29–39, 2014. ISSN 0376-0421. URL `https://www.sciencedirect.com/science/article/pii/S0376042114000426`.

Börjesson et al., **2022**. Elias Börjesson, Joris J.C. Remmers and Martin Fagerström. *A generalised path-following solver for robust analysis of material failure.* PHD, 2022.

Camanho et al., **2003**. P. P Camanho, C. G Davila and M. F de Moura. *Numerical Simulation of Mixed-Mode Progressive Delamination in Composite Materials.* Journal of composite materials, 37(16), 1415–1438, 2003. ISSN 0021-9983.

Cheng et al., **2019**. Chao Cheng, Chenyu Zhang, Jinli Zhou, Minqiang Jiang, Zeyu Sun, Shuai Zhou, Yong Liu, Zhengguo Chen, Lei Xu, Hui Zhang and Muhuo Yu. *Improving the interlaminar toughness of the carbon fiber/epoxy composites via interleaved with polyethersulfone porous films.* Composites Science and Technology, 183, 107827, 2019. ISSN 0266-3538. doi: https://doi.org/10.1016/j.compscitech.2019.107827. URL `https://www.sciencedirect.com/science/article/pii/S0266353819313806`.

Crisfield, **2000**. M.A. Crisfield. *Non-linear finite element analysis of solids and structures.*, 2000.

de Borst et al., **2012**. René de Borst, Mike A Crisfield, Joris J.C Remmers and Clemens V Verhoosel. *Non-Linear Finite Element Analysis of Solids and Structures: Second Edition.* John Wiley & Sons Ltd, Chichester, West Sussex, U.K, 2nd ed. edition, 2012. ISBN 0470666447.

Falk et al., **2001**. M. L. Falk, A. Needleman and J. R. Rice. *A critical evaluation of cohesive zone models of dynamic fractur.* J. Phys. IV France, 11, 43–50, 2001. doi: 10.1051/jp4:2001506. URL `https://doi.org/10.1051/jp4:2001506`.

Geers, **1999**. M. G. D Geers. *Enhanced solution control for physically and geometrically non-linear problems. Part I-the subplane control approach.* International journal for numerical methods in engineering, 46(2), 177–204, 1999. ISSN 0029-5981.

Gnaba et al., **2019**. I Gnaba, X Legrand, P Wang and D Soulat. *Through-the-thickness reinforcement for composite structures: A review.* Journal of Industrial Textiles, 49, 71–96, 2019.

Goutianos and Sørensen, **2016**. Stergios Goutianos and Bent F Sørensen. *Fracture resistance enhancement of layered structures by multiple cracks.* Engineering fracture mechanics, 151, 92–108, 2016. ISSN 0013-7944.

Goyal, **2002**. Vinay K Goyal. *Analytical modeling of the mechanics of nucleation and growth of cracks.* 2002.

Gutierrez, **2004**. MA Gutierrez. *Energy release control for numerical simulations of failure in quasi-brittle solids.* Communications in numerical methods in engineering, 20 (1), 19–29, 2004. ISSN 1069-8299.

Harper and Hallett, **2008**. Paul W. Harper and Stephen R. Hallett. *Cohesive zone length in numerical simulations of composite delamination.* Engineering Fracture Mechanics, 75(16), 4774–4792, 2008. ISSN 0013-7944. doi: https://doi.org/10.1016/j.engfracmech.2008.06.004. URL `https://www.sciencedirect.com/science/article/pii/S0013794408001653`.

Heß and Himmel, **2010**. H. Heß and N. Himmel. *Structurally stitched NCF CFRP laminates. Part 1: Experimental characterization of in-plane and out-of-plane properties.* Composites Science and Technology, 71, Pages 549–568, 2010.

Hillerborg et al., **11 1976**. A. Hillerborg, M. Modéer and P.-E Petersson. *Analysis of Crack Formation and Crack Growth by Means of Fracture Mechanics and Finite Elements.* Cement and Concrete Research, 6, 773–781, 1976. doi: 10.1016/0008-8846(76)90007-7.

Jensen et al., **2018**. S. M. Jensen, M. Martos Sanchez, B. L. V. Bak and E. Lindgaard. *Identification of a novel cohesive zone law in an interface finite element for simulating delamination with R-curve effects.* In ECCM 2018 - 18th European Conference on Composite Materials, 2018.

Jensen, **2021**. Simon Mosbjerg Jensen. *Delamination in composites with fiber bridging under quasi-static loading and variable amplitude fatigue loading*, 2021. ISSN 24461636.

Jensen et al., **2019**. Simon Mosbjerg Jensen, Mario Martos Sanchez, Brian Lau Verndal Bak and Esben Lindgaard. *Formulation of a mixed-mode multilinear cohesive zone law in an interface finite element for modelling delamination with R-curve effects.* Composite Structures, 216, 477–486, 2019. ISSN 0263-8223. doi: https://doi.org/10.1016/j.compstruct.2019.02.029. URL `https://www.sciencedirect.com/science/article/pii/S0263822318337383`.

Jones, **1999**. Robert M. Jones. *Mechanics of composite materials.* Brunner-Routledge, 2. ed. edition, 1999. ISBN 1-56032-712-X.

Leon et al., **2011**. S.E Leon, G. H. Paulino, Anderson Pereira, I.F.M. Menezes and E.N. Lages. *A Unified Library of Nonlinear Solution Schemes.* Applied Mechanics Reviews, 64, 2011.

Lindgaard et al., **2017**. E. Lindgaard, B.L.V. Bak, J.A. Glud, J. Sjølund and E.T. Christensen. *A user programmed cohesive zone finite element for ANSYS Mechanical.* Engineering Fracture Mechanics, 180, 229–239, 2017. ISSN 0013-7944. doi: https://doi.org/10.1016/j.engfracmech.2017.05.026.

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, 220, 953–960, 2019.

MathWorks, **2022a**. MathWorks. *Computer Vision Toolbox Documentation.* `https://se.mathworks.com/help/vision/index.html?s_tid=CRUX_lftnav`, 2022a. Accessed: 11-4-2022.

MathWorks, **2022b**. MathWorks. *Local Feature Detection and Extraction.* `https://se.mathworks.com/help/vision/ug/local-feature-detection-and-extraction.html`, 2022b. Downloadet: 11-4-2022.

McGugan et al., **2015**. M McGugan, G Pereira, B.F Sørensen, H Toftegaard and K Branner. *Damage tolerance and structural monitoring for wind turbine blades.* Philosophical topics, 373(2035), 20140077–, 2015. ISSN 0276-2080.

Mikolajczyk and Tuytelaars, **2009**. Krystian Mikolajczyk and Tinne Tuytelaars. *Local Image Features.* Springer US, Boston, MA, 2009. ISBN 978-0-387-73003-5. doi: 10.1007/978-0-387-73003-5_224. URL `https://doi.org/10.1007/978-0-387-73003-5_224`.

Palazzetti and Zucchelli, **2017**. R. Palazzetti and A. Zucchelli. *Electrospun nanofibers as reinforcement for composite laminates materials – A review.* Composite Structures, 182, 711–727, 2017. ISSN 0263-8223. doi: https://doi.org/10.1016/j.compstruct.2017.09.021. URL `https://www.sciencedirect.com/science/article/pii/S0263822317318196`.

PRO-SET inc., **2022**. PRO-SET inc. *Technical Data INF-114 INF-212 Infusion epoxy.* `https://www.prosetepoxy.com/wp-content/uploads/INF-114_INF-212.pdf`, 2022. Online; accessed 22 March 2022.

Rask and Sørensen, **2012**. M Rask and B.F Sørensen. *Determination of the J integral for laminated double cantilever beam specimens: The curvature approach.* Engineering fracture mechanics, 96, 37–48, 2012. ISSN 0013-7944.

Shivakumar and Panduranga, **2013**. Kunigal N. Shivakumar and Raghu Panduranga. *Interleaved Polymer Matrix Composites - A Review.* In: 54th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference, page 1903, 2013.

Sun and Jin, **2012**. Chin-Teh Sun and Zhihe Jin. *Fracture Mechanics.* Academic Press, 2012. ISBN 9780123850010.

Sørensen et al., 2006. Bent F. Sørensen, Kenneth Jørgensen, Torben K. Jacobsen and Rasmus C. Østergaard. *DCB-specimen loaded with uneven bending moments.* International journal of fracture, 141, 163–176, 2006. ISSN 0376-9429.

Tavares and de Castro, 2018. Sérgio M. O Tavares and Paulo M. S. T de Castro. *Damage Tolerance of Metallic Aircraft Structures: Materials and Numerical Modelling.* Springer International Publishing AG, Cham, 2018. ISBN 3319701894.

Trabal et al., 2022a. Guillem Gall Trabal, Brian Lau Verndal Bak, Boyang Chen, Simon Mosbjerg Jensen and Esben Lindgaard. *Multiple Delamination Toughening of Composite Laminates Using Weakening or Toughening Interlaminar Patches: A Numerical Study.* (Under Review), 2022a.

Trabal et al., 2022b. Guillem Gall Trabal, Brian Lau Verndal Bak, Boyang Chen and Esben Lindgaard. *An adaptive floating node based formulation for the analysis of multiple delaminations under quasi-static loading.* Composites. Part A, Applied science and manufacturing, 156, 2022b. ISSN 1359-835X.

Turon et al., 2006. A Turon, P.P Camanho, J Costa and C.G Dávila. *A damage model for the simulation of delamination in advanced composites under variable-mode loading.* Mechanics of materials, 38(11), 1072–1089, 2006.

Turon et al., 2007. A. Turon, C.G. Dávila, P.P. Camanho and J. Costa. *An engineering solution for mesh size effects in the simulation of delamination using cohesive zone models.* Engineering Fracture Mechanics, 74(10), 1665–1682, 2007. ISSN 0013-7944. doi: https://doi.org/10.1016/j.engfracmech.2006.08.025. URL `https://www.sciencedirect.com/science/article/pii/S0013794406003808`.

Turon et al., 2010. A. Turon, P.P. Camanho, J. Costa and J. Renart. *Accurate simulation of delamination growth under mixed-mode loading using cohesive elements: Definition of interlaminar strengths and elastic stiffness.* Composite Structures, 92(8), 1857–1864, 2010. ISSN 0263-8223. doi: https://doi.org/10.1016/j.compstruct.2010.01.012.

Turon, 2007. Albert Turon. *Simulation of delamination in composites under quasi-static and fatigue loading using cohesive zone models*, 2007.

Vasios, 2015. Nikolaos Vasios. *Nonlinear Analysis of Structures*, 2015.

Verhoosel et al., 2009. Clemens V Verhoosel, Joris J. C Remmers and Miguel A Gutiérrez. *A dissipation-based arc-length method for robust simulation of brittle and ductile failure.* International journal for numerical methods in engineering, 77(9), 1290–1321, 2009. ISSN 0029-5981.

Xu et al., 2018. Feng Xu, De dong Huang and Xusheng Dub. *Improving the delamination resistance of carbon fiber/epoxy composites by brushing and abrading of the woven fabrics.* Construction and Building Materials, 158, 257–263, 2018.

Yang et al., 2006. Q.D. Yang, Brian N. Cox, Ravi K. Nalla and R.O. Ritchie. *Fracture length scales in human cortical bone: The necessity of nonlinear fracture models.* Biomaterials, 27(9), 2095–2113, 2006. ISSN 0142-9612. doi:

https://doi.org/10.1016/j.biomaterials.2005.09.040. URL https://www.sciencedirect.com/science/article/pii/S0142961205008720.

Yang and Cox, **01 2005**. Qingda Yang and B. Cox. *Cohesive models for damage evolution in laminated composites.* International Journal of Fracture, 133, 107–137, 2005. doi: 10.1007/s10704-005-4729-6.

Yasaee et al., **2012**. M. Yasaee, I.P. Bond, R.S. Trask and E.S. Greenhalgh. *Mode I interfacial toughening through discontinuous interleaves for damage suppression and control.* Composites Part A: Applied Science and Manufacturing, 43, 198–207, 2012.

İnal O. et al., **2022**. İnal O., Katnam K.B., Potluri P. and Soutis C. *Progress in interlaminar toughening of aerospace polymer composites using particles and non-woven veils.* Aeronautical Journal, 126(1295), 222 – 248, 2022. doi: 10.1017/aer.2021.95.

Cohesive Element used in the MATLAB Framework

A

The following appendix outlines the cohesive element described in Turon et al. (2006). This element is used in the MATLAB Framework to model delamination. Further details on the cohesive element can be found in (Turon et al., 2006). In its original formulation, the element is a 3D, 8-noded cohesive element. However in this work, the element is used in a 2D configuration Trabal et al. (2022b). Thus, mode III crack opening is not considered.

Kinematic formulation: In this element the displacement discontinuity is described by an upper crack surface S^+ and a lower crack surface S^-. For an undamaged interface the element has zero thickness and the upper and lower crack surface are coincident. Upon opening of the interface, the upper crack surface, S^+, moves a distance $\{\boldsymbol{u}^+\}$ and the lower crack surface, S^-, a distance $\{\boldsymbol{u}^-\}$. Thus, the separation in global coordinates, $\{\boldsymbol{s}_G\}$, are defined by Equation A.1.

$$\{\boldsymbol{s}_G\} = \{\boldsymbol{u}^+\} - \{\boldsymbol{u}^-\} \tag{A.1}$$

The separation is evaluated at the interface midsurface, $\{\bar{\boldsymbol{x}}\}$, which is defined in global coordinates in Equation A.2.

$$\{\bar{\boldsymbol{x}}\} = \{\boldsymbol{X}\} + \frac{1}{2}(\{\boldsymbol{u}^+\} + \{\boldsymbol{u}^-\}) \tag{A.2}$$

$\{\bar{\boldsymbol{x}}\}$ is the coordinate of the midsurface in the deformed configuration and $\{\boldsymbol{X}\}$ is the initial coordinates in the undeformed configuration.

Further, in order to distinguish between the different opening modes the separation must be described in a local coordinate system. This is done by establishing a rotation matrix, $[\boldsymbol{T}]$, which relates the orientation of the interface midsurface, in local coordinates to the global coordinates.

Utilizing the rotation matrix, the separation in the local coordinates, $\{\boldsymbol{s}_L\}$, is defined by Equation A.3.

$$\{\boldsymbol{s}_L\} = [\boldsymbol{T}](\{\boldsymbol{u}^+\} - \{\boldsymbol{u}^-\}) \tag{A.3}$$

Constitutive law: The constitutive law used in the cohesive element is a bilinear traction-separation law, as illustrated in Figure A.1. The separation in local coordinates, $\{\boldsymbol{s}_L\}$, is related to the tractions, $\{\boldsymbol{\tau}\}$, by Equation A.4.

$$\{\boldsymbol{\tau}\} = (1-d)[\boldsymbol{I}]E_p\{\boldsymbol{s}_L\} = (1-d)[\boldsymbol{E}_p]\{\boldsymbol{s}_L\} \tag{A.4}$$

Where d, is a scalar damage parameter and defines the amount of damage in the interface. This parameter has a value between 1 and 0, where 1 indicates a fully damaged interface and 0 indicates an undamaged interface.

E_p, in Equation A.4, is the penalty stiffness of the undamaged cohesive interface, and defines the slope of the initial linear part of the curve, prior to damage. The penalty stiffness is used to maintain compatibility in undamaged interface before crack onset. However, the introduction of the penalty stiffness generally yields a more compliant interface. Thus, the penalty stiffness must be large enough to guarantee a stiff connection in the interface prior to damage, such that the added compliance is insignificant. However, the penalty stiffness should also be chosen small enough to avoid numerical problems in the solution. (Bak, 2015)

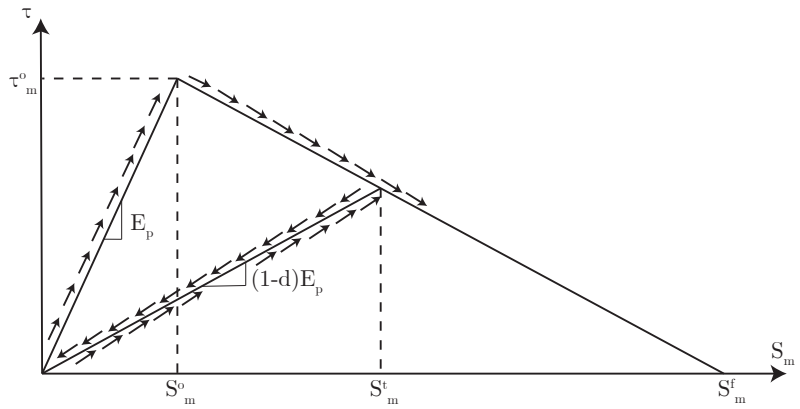


Figure A.1. *Bilinear traction-separation law for mixed mode loading used in the cohesive element. τ_m^o is the onset traction, S_m^o is the onset separation norm, S_m^t is the current threshold separation norm, S_m^f is the critical separation norm, E_p is the penalty stiffness and, d, is the damage parameter.*

The traction-seperation law in Equation A.4 can be applied if the damage parameter, d, is known at all times during the damage process. In order to describe, d, throughout the damage process, a damage evolution law is needed. The damage evolution law it defined in terms of a damage initiation criterion and a damage propagation criterion. These are explained in the following.

As explained in section 1.2, delamination cracks are prone to grow under mixed mode conditions. Due to this the cohesive element must be formulated in a way, that accounts for mixed mode conditions. This is done through the definition of an equivalent one dimensional separation norm, S_m, the damage initiation criterion, and the damage propagation criterion. In order to take mixed mode into account when formulating the damage initiation criterion and the damage propagation cirterion, a mixed mode ratio is determined as in Equation A.5.

$$\beta = \frac{s_{II}}{\langle s_I \rangle + s_{II}} \tag{A.5}$$

Where s_I and s_{II} are the separation in mode I and II, respectively. Furthermore, the mode I separation is defined as $\langle s_I \rangle = \frac{1}{2}(|s_I| + s_I)$ in order to avoid penetration in the interface, by always ensuring a non-negative mode I opening.

Norm of separation: The separation norm, S_m, is defined in Equation A.6, as an equivalent one dimensional opening displacement for a mixed mode opening.

$$S_m = \sqrt{\langle s_I \rangle^2 + s_{II}^2} \tag{A.6}$$

Where the same definition of $\langle s_I \rangle$ as in Equation A.5 is used.

Damage evolution law: The evolution criteria for the damage parameter, d, is defined in Equation A.7 where, d is the current damage parameter and, r is a damage threshold parameter (Bak, 2015).

$$d - r \leq 0 \tag{A.7}$$

Thus, the condition, $d - r < 0$ implies that the current damage parameter, d is below the threshold value, r, and the damage state remains unchanged. On the other hand, the condition $d - r = 0$ implies that damage may evolve.

The evolution law for the damage parameter, d, is defined in terms of the separation norm, S_m, as done in Equation A.8 for the bilinear traction-separation law.

$$d(S_m) = \frac{S_m^f(S_m - S_m^o)}{S_m(S_m^f - S_m^o)} \qquad \text{d} \in [0,1] \quad \text{for } S_m \in [S_m^o, S_m^f] \tag{A.8}$$

where S_m^o is the onset separation norm, and S_m^f is the critical separation norm. Both, S_m^o and S_m^f are defined in terms of the mixed mode ratio, β. The onset separation norm, S_m^o, is determined based on the damage initiation law, while the critical separation norm, S_m^f, is obtained from the damage propagation law.

With offset in the evolution law defined in Equation A.8, the damage threshold value, r is defined in Equation A.9

$$d(S_m^t) = r = \frac{S_m^f(S_m^t - S_m^o)}{S_m^t(S_m^f - S_m^o)} \tag{A.9}$$

Where S_m^t is the threshold separation norm, S_m^t, illustrated in Figure A.1.

Based on the evolution criteria in Equation A.7 and the definition of the damage threshold value, r, in Equation A.9, it is possible to define the threshold separation norm, S_m^t as in Equation A.10

$$S_m^t = \frac{S_m^o S_m^c}{S_m^c - d(S_m^c - S_m^o)} \tag{A.10}$$

A change in the damage parameter requires the separation norm, S_m, to exceed the current threshold separation norm, S_m^t, as illustrated in Figure A.1. For a pristine interface, the threshold separation norm, S_m^t, is equal to the onset separation norm, S_m^o. Thereby, damage initiation occurs once the separation norm, S_m exceeds S_m^o.

Furthermore, the damage parameter can only increase, as damage is irreversible. This means the damage parameter for the next increment, $n + 1$, is chosen according to Equation A.11

$$d^{n+1} = \min\big(\max(d(S_m),r^n),1\big) \tag{A.11}$$

where, r^n is the current damage threshold value.

Damage propagation: Crack propagation occurs, when the energy release rate, G, is greater than or equal to the critical energy release rate, G_c. In a mixed mode crack configuration, the energy release rate is determined by:

$$G = G_I + G_{II} \tag{A.12}$$

where G_I and G_{II} are the energy release in mode I and mode II, respectively.

The critical energy release rate is determined by a Benzeggagh-Kenane (BK) criterion and defined in Equation A.13.

$$G_c = G_{Ic} + (G_{IIc} - G_{Ic})B(\beta)^\eta \tag{A.13}$$

Where η is a mode interaction parameter determined from experimental data and $B(\beta)$ is defined in Equation A.14.

$$B(\beta) = \frac{G_{II}}{G_I + G_{II}} = \frac{\beta^2}{1 + 2\beta^2 - 2\beta} \tag{A.14}$$

The propagation criteria can further be written in the separation space by utilizing, that the critical energy release rate for the bilinear traction-separation law is defined by $G_c = \frac{1}{2}E_p S_m^o S_m^f$. Utilizing this, the critical separation norm, S_m^f, is defined by Equation A.15.

$$S_m^f = \frac{s_I^o s_I^f + (s_{II}^o s_{II}^f - s_I^o s_I^f)B(\beta)^\eta}{S_m^o} \tag{A.15}$$

Damage initiation criterion: As explained above, damage initiation occurs when the traction in the interface exceeds its critical value. The mixed mode traction value is defined according to Equation A.16.

$$\tau = \langle \tau_I \rangle^2 + \tau_{II}^2 \tag{A.16}$$

In order to predict damage initiation in a mixed mode configuration, the value of the onset traction, τ^o, must also take the mixed mode condition into account. This is done by utilizing an approach similar to the damage propagation criteria. Thus, an equivalent one dimensional onset traction, τ^o, is also determined by a Benzeggagh-Kenane (BK) criterion. The exact definition of the onset traction is defined in Equation A.17

$$\tau^o(\beta)^2 = \tau_I^{o2} + (\tau_{II}^{o^2} - \tau_I^{o^2})B(\beta)^\eta \tag{A.17}$$

Where $B(\beta)^\eta$ is equal to Equation A.14. Using the bilinear traction-separation law, the separation norm at damage initiation, S_m^o, is defined by

$$(S_m^o)^2 = (s_I^o)^2 + \big((s_{II}^o)^2 - (s_I^o)^2\big)B(\beta)^\eta \tag{A.18}$$

Riks Arc-length Solver B

In this appendix, the implementation of Riks arc-length solver in the MATLAB Framework is explained and validated. This Arc-length solver is implemented, as it uses the same structure of equations as the two energy based arc-length solver explained in chapter 5. However, this method is simpler and can be testes against the simple 1D and 2D benchmark examples in section 4.2. Due to these reasons this algorithm is easier to debug in case mistakes are made.

Thereby, this arc-length solver is implemented with the main purpose of validating that the structure of the equations in the solver are correct. When this has been successfully done, it is possible to implement the two energy based arc-length solver by changing the path-following constraint and its derivatives. Thereby, the intent is to minimize and narrow down potential bugs in the energy based arc-length solvers. This is especially useful, when implemented in the MATLAB Framework, where the floating node method based code complicates the implementation. In the following the equations used in the Riks arc-length solver are first defined and finally the method is validated against the problem in section 4.2.

Riks arc-length solver is based on the same set of equations as the Crisfield arc-length solver explained in section 4.1. This means an augmented system of equations is used as defined in Equation 4.6 and repeated below to ease the reading.

$$\begin{Bmatrix} \{\boldsymbol{F}_{int}(\{\boldsymbol{D}\})\}^n - \lambda^n\{\boldsymbol{F}_{ext}\} \\ g^n \end{Bmatrix} = \begin{Bmatrix} \{\boldsymbol{0}\} \\ 0 \end{Bmatrix} \tag{4.6}$$

This set of equations is solved by linearising the equation by truncated Taylor series expansions as stated in Equation 4.7. Again this equation is repeated below to ease the reading.

$$\begin{Bmatrix} \{\boldsymbol{F}_{int}\}_i^n - \lambda_i^n\{\hat{\boldsymbol{F}}_{ext}\} + \frac{\partial\{\boldsymbol{F}_{int}\}}{\partial\{\boldsymbol{D}\}}\{\delta\boldsymbol{D}\}_i^n - \{\hat{\boldsymbol{F}}_{ext}\}\delta\lambda_i^n \\ g_i^n + \left(\frac{\partial g}{\partial\{\boldsymbol{D}\}}\right)^T\{\delta\boldsymbol{D}\}_i^n + \frac{\partial g}{\partial\lambda}\delta\lambda_i^n \end{Bmatrix} = \begin{Bmatrix} \{\boldsymbol{0}\} \\ 0 \end{Bmatrix} \tag{4.7}$$

Rewriting this equation leads to the equation in Riks arc-length solver, which is used to calculate corrections to the displacement field and load-level parameter. This equation is stated in Equation 4.8 and repeated below.

$$\begin{bmatrix} [\boldsymbol{K}_t]_i^n & -\{\hat{\boldsymbol{F}}_{ext}\} \\ (\{\boldsymbol{h}\}_i^n)^T & w_i^n \end{bmatrix} \begin{Bmatrix} \{\delta\boldsymbol{D}\}_i^n \\ \delta\lambda_i^n \end{Bmatrix} = \begin{Bmatrix} \{\boldsymbol{R}\}_i^n \\ -g_i^n \end{Bmatrix} \tag{4.8}$$

Where the tangent stiffness, $[\boldsymbol{K}_t]$, a h-vector, $\{\boldsymbol{h}\}$ and scalar, w, is defined in Equation 4.9.

$$[\boldsymbol{K}_t]_i^n = \frac{\partial\{\boldsymbol{F}_{int}\}}{\partial\{\boldsymbol{D}\}} \qquad \{\boldsymbol{h}\}_i^n = \frac{\partial g}{\partial\{\boldsymbol{D}\}} \qquad w_i^n = \frac{\partial g}{\partial\lambda} \tag{4.9}$$

The path-following constraint is the same as the one that is used in Crisfield arc-length solver defined in Equation 4.16. This path-following constraint is repeated below:

$$g = \left(\{\Delta \boldsymbol{D}\}_i^n\right)^T \{\Delta \boldsymbol{D}\}_i^n + \beta^2 (\Delta\lambda_i^n)^2 \{\hat{\boldsymbol{F}}_{ext}\}^T \{\hat{\boldsymbol{F}}_{ext}\} - \Delta l^2 = 0 \tag{4.16}$$

However, in Riks arc-length solver this constraint is introduced in a linearised form according to Equation 4.8. This means it is necessary to define expressions for the h-vector and the scalar, w, defined in Equation 4.9. Based on the definition of the path-following constraint in Equation 4.16 the h-vector and scalar, w is defined in Equation B.1.

$$\{\boldsymbol{h}\}_i^n = \frac{\partial g}{\partial \{\boldsymbol{D}\}} = 2(\{\Delta \boldsymbol{D}\}_i^n)^T \qquad w = \frac{\partial g}{\partial \lambda} = 2\beta^2 \lambda_i^n \{\hat{\boldsymbol{F}}_{ext}\}^T \{\hat{\boldsymbol{F}}_{ext}\} \tag{B.1}$$

After corrections to the load level parameter, $\delta\lambda$, and displacement field, $\{\delta \boldsymbol{D}\}$ is found from Equation 4.8, the total incremental change in the load-level parameter and displacement field are updated according to Equation 4.14 and 4.15. The updating procedure for the total incremental change in the load-level parameter and displacement field, from Equation 4.14 and 4.15 are further repeated below.

$$\{\Delta \boldsymbol{D}\}_{i+1}^n = \{\Delta \boldsymbol{D}\}_i^n + \{\delta \boldsymbol{D}\}_i^n \tag{4.14}$$

$$\Delta\lambda_{i+1}^n = \Delta\lambda_i^n + \delta\lambda_i^n \tag{4.15}$$

The predictor step used in the implemented arc-length solver is equivalent to the predictor step used in Crisfield arc-length solver in subsection 4.1.1. This predictor step is used as it seemed to work fine in the Crisfield arc-length solver.

B.1 Validation of Riks Arc-Length Solver

The validation of Riks arc-length solver is done on the same example as is used in section 4.2 and the results are illustrated in Figure B.1. This problem is again chosen as it features both a snap-through and snap-back behaviour. The same problem is run with different solver settings, i.e. residual and arc-lengths to test the robustness of the solver. This is done as the authors have experienced, that bugs in the codes can introduce convergence problems for specific solver settings.

In this figure each substep is highlighted with a marker according to the legend. In addition to the tests illustrated in Figure B.1 a test with an arc-length of, $\Delta l = 0.0001$ has also been done. However this is not plotted in the figure as the small distance between each substep makes it difficult to see the remaining tests.

From figure Figure B.1 it is clear that Riks arc-length solver can solve past the snap-through in point 1 and the snap-back in point 2. However, in order for the solver to advance past point 1 and 2 a sufficiently small step sizes is needed. In the test runs illustrated in Figure B.1, the arc-length is kept constant throughout the solution. For an arc-length of $\Delta l = 0.005$ the solver fails at point 1. For an arc-length of $\Delta l = 0.001$ the solver fails at point 2, whereas the solver is able to trace the full curve for an arc-length of $\Delta l = 0.0005$. In order to ensure a sufficiently small step size, a condition which halves the arc-length if the number of iterations or the residual exceeds a predefined value.

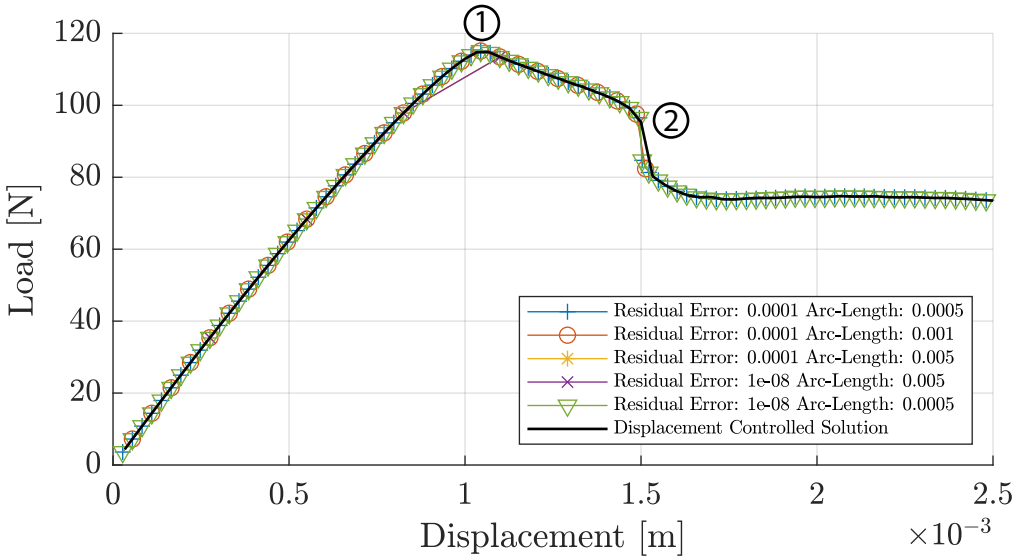


Figure B.1. *Validation of Riks arc-length solver. All test are run with* $\beta = 0$.

Based on the test runs in it is assessed that the Riks arc-length solver works satisfactory, as it traces the displacement controlled solution almost perfectly.

An algorithm for the implementation of Riks arc-length solver in the MATLAB Framework is shown in 2.

Algorithm 2 Riks arc-length solver

Save analysis data from last converged substep into new variables
Initialize vectors for $\{\delta \boldsymbol{D}^I\}$, $\{\delta \boldsymbol{D}\}$, $\{\Delta \boldsymbol{D}\}$ and scalars for $\delta\lambda$ and $\Delta\lambda$
Initialize iteration counter
if Substep = 1 **then**
    Update refinement of A-FNM elements
    Calculate and assemble the stiffness matrix and internal force vector
end if
Calculate displacement increment $\{\delta \boldsymbol{D}^I\}$ according to Equation 4.13
Calculate predictor value for load-level parameter $\Delta\lambda_1^n$ according to Equation 4.26
Calculate predictor displacement increment $\{\Delta \boldsymbol{D}\}_1$ according to Equation 4.22
Update displacement $\{\boldsymbol{D}\}_1^n = \{\boldsymbol{D}\}^{n-1} + \{\delta \boldsymbol{D}\}_1^n$
Update load-level parameter $\lambda_1^n = \lambda^{n-1} + \delta\lambda_1^n$
Update refinement of A-FNM elements
Calculate and assemble the stiffness matrix and internal force vector
Evaluate the residual force vector according to Equation 4.2
Update MPCs in the A-FNM elements
while The norm of the residual force is larger than the residual tolerance **do**
    Update iteration counter
    Calculate the constraint derivatives, $\{\boldsymbol{h}\}$ and, w, according to Equation B.1
    Calculate the path-following constraint according to Equation 4.16
    Assemble the system of equations defined in Equation 4.8 and solve for the load-level parameter correction $\delta\lambda_i^n$ and the displacement correction $\{\delta \boldsymbol{D}\}_i^n$
    Update the incremental load-level parameter $\Delta\lambda$ according to Equation 4.15
    Update the incremental displacement vector $\{\Delta \boldsymbol{D}\}$ according to Equation 4.14
    Update displacement $\{\boldsymbol{D}\}_{i+1}^n = \{\boldsymbol{D}\}^{n-1} + \{\Delta \boldsymbol{D}\}_{i+1}^n$
    Update load-level parameter $\lambda_{i+1}^n = \lambda^{n-1} + \Delta\lambda_{i+1}^n$
    Update refinement of A-FNM elements
    Calculate and assemble the stiffness matrix and internal force vector
    Evaluate the residual force vector according to Equation 4.2
    if Residual error or number or iterations exceeds a predefined value **then**
        Reset all analysis data to last converged substep,
        Bisect the step size, Δl,
        Exit solver
    end if
    Update MPCs in the A-FNM elements
end while
Save all analysis data
Exist Riks arc-length solver

Validation of Crisfield Arc-Length Method on Simple Benchmark Examples

C

In this section the programmed arc-length solver is validated against three benchmarks examples. This is done in parallel with the development of the arc-length solver, in order to gradually increase the complexity, while also verifying it along the way. First a 1 DOF system is considered and afterwards two 2 DOF's benchmarks examples are used to validate the solver. Finally the arc-length solver is implemented in the MATLAB Framework and validated against a displacement controlled solution.

C.0.1 One Degree of Freedom System:

A 1 DOF benchmark example is used as the first and simplest benchmark example. This example is implemented as it makes it easy to plot how the solution progresses, thus making debugging easier. In this benchmark example the arc-length solver is set up to trace the polynomial defined in Equation C.1. This polynomial is chosen as it models a snap-through behaviour, which is generally not trivial to pass with a non-linear solver.

$$f(x) = 0.633x^3 - 5x^2 + 10x \tag{C.1}$$

The full solution plotted on top of the analytical function is illustrated in Figure C.1.

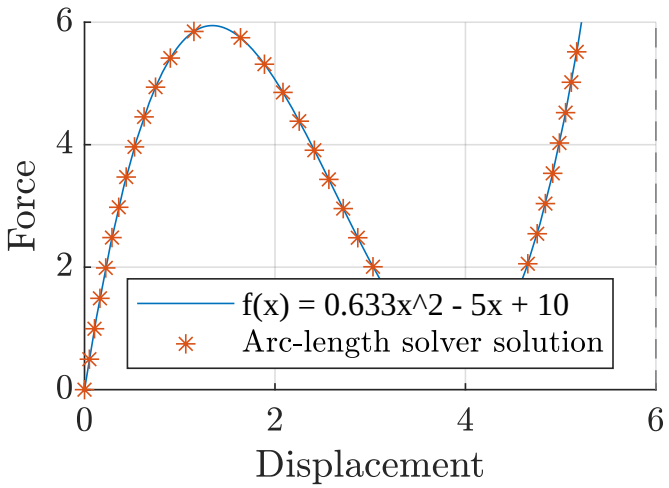


Figure C.1. *Arc-length solver solution for 1DOF system. Solver settings are* $\Delta l = 0.5$, $\beta = 1$, *Residual tolerance = 0.0001*

Additionally, several solver settings are tested to validate the robustness of the solver. The result from each of these test, along with the specific solver settings are listed in Table C.1. Based on the tests it is deemed that the arc-length solver works satisfactory.

Test:	Δl	β	Residual Tolerance	Converged	Substeps	Total Iterations
Test 1	0.0001	1	$1.0 \cdot 10^{-4}$	Yes	191947	4
Test 2	0.001	1	$1.0 \cdot 10^{-4}$	Yes	19195	48
Test 3	0.01	1	$1.0 \cdot 10^{-4}$	Yes	1920	359
Test 4	0.1	1	$1.0 \cdot 10^{-4}$	Yes	192	195
Test 5	1.0	1	$1.0 \cdot 10^{-4}$	Yes	19	38
Test 6	5	1	$1.0 \cdot 10^{-4}$	Yes	5	13
Test 7	0.0001	0	$1.0 \cdot 10^{-4}$	Yes	53202	12
Test 8	0.001	0	$1.0 \cdot 10^{-4}$	Yes	5321	130
Test 9	0.01	0	$1.0 \cdot 10^{-4}$	Yes	533	464
Test 10	0.1	0	$1.0 \cdot 10^{-4}$	Yes	54	53
Test 11	1.0	0	$1.0 \cdot 10^{-4}$	Yes	6	6
Test 12	5	0	$1.0 \cdot 10^{-4}$	Yes	2	2
Test 13	0.5	1	$1.0 \cdot 10^{-10}$	Yes	38	101
Test 14	0.5	0	$1.0 \cdot 10^{-10}$	Yes	11	11

Table C.1. *Robustness check of the arc-length solver.*

C.0.2 Two Degree of Freedom System:

In the previous 1 DOF benchmark, only scalar values were used. However, in order to solve the benchmark examples with 2 DOF the code must be generalised to handle matrices and vectors. This is also needed later when the solver is implemented in the MATLAB Framework. The two benchmark examples in this section are further chosen, as these both includes snap-through and snap-back behaviour.

Benchmark One

The first of the two, 2 DOF benchmark examples takes offset in the truss structure illustrated in Figure C.2 and is based on the article by Vasios (2015). It is assumed that the truss members cannot buckle, and are only allowed to deform by compression or extension. It is further assumed that each truss member are homogeneous and made of a isotropic, linearly elastic material. u_1 and u_2 are the vertical displacement of the hinge points and a load, F is applied. All hinges in the structure are constrained to only allow vertical displacement. l_0 and L_0 defines the initial length of the truss members, while β and k define their stiffness.

The equilibrium equations for this system are defined in Equation C.2 and details of the derivation can be found in Vasios (2015).

$$\begin{aligned} \left(\frac{1}{\sqrt{B(a_1,\theta_0)}} - 1\right)(\sin(\theta_0) - a_1) - w(a_2 - a_1) &= 0 \\ w(a_2 - a_1) - \lambda &= 0 \end{aligned} \tag{C.2}$$

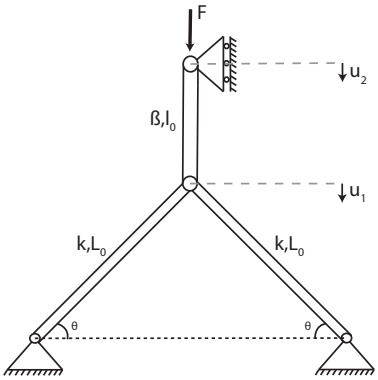


Figure C.2. *Truss structure for 2 Degree of freedom benchmark example.*

$a_2 = \frac{u_2}{L_0}$ and $a_1 = \frac{u_1}{L_0}$ are normalized displacements and $\lambda = \frac{P}{2kL}$ is the normalized force. $B(a_1,\theta_0)$ is defined in Equation C.3 and $w = \frac{\beta}{k}$, defines the stiffness ratio between the individual truss members.

$$B(a_1,\theta_0) = 1 - 2a_1\sin(\theta_0) + a_1^2 \tag{C.3}$$

The system of equations in Equation C.2 can be expressed in residual form, as defined in Equation C.4.

$$\boldsymbol{f}_{int}(\boldsymbol{a}) - \lambda \boldsymbol{q} = \begin{bmatrix} \left(\frac{1}{\sqrt{B(a_1,\theta_0)}} - 1\right)(\sin(\theta_0) - a_1) - w(a_2 - a_1) \\ w(a_2 - a_1) \end{bmatrix} - \lambda \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{C.4}$$

Differentiating $\boldsymbol{f}_{int}$ in Equation C.4 with respect to both a_1 and a_2 yields the tangent stiffness matrix for the system.

It is not possible to determine an analytical solution for the benchmark example. The programmed arc-length solver is therefore validated by visually comparing the solution from the developed arc-length solver with the solution from Vasios (2015).

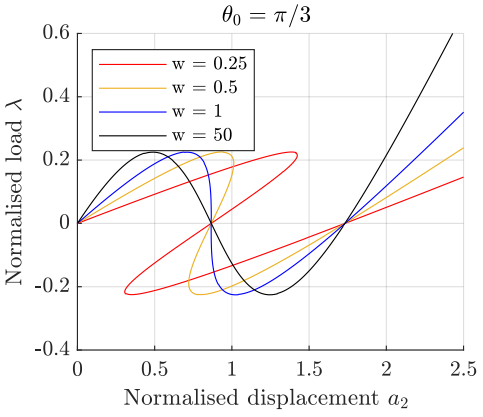


Figure C.3. *Arc-length solver solution for 2DOF system. Solver settings are* $\Delta l = 0.005$, $\beta = 0$, *Residual tolerance = 0.001.*

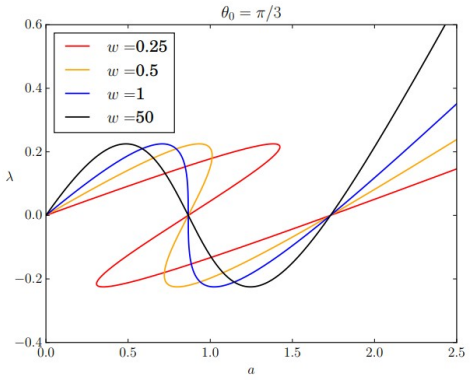


Figure C.4. *Benchmark example with normalised load* λ *plotted as a function of the normalised displacement* a_2 *based. (Vasios, 2015)*

Based on the results in Figure C.3 and C.4, the programmed arc-length solver is deemed to work, when snap-back behaviour occurs. Again, several solver settings are tested to

validate the robustness of the solver. The result from each test, along with the specific solver settings are listed in Table C.2.

Test:	Δl	β	Residual Tolerance	Converged	Substeps	Total Iterations
Test 1	$0.5 \cdot 10^{-4}$	1	$1.0 \cdot 10^{-6}$	Yes	197280	47
Test 2	$0.5 \cdot 10^{-3}$	1	$1.0 \cdot 10^{-6}$	Yes	19728	451
Test 3	$0.5 \cdot 10^{-2}$	1	$1.0 \cdot 10^{-6}$	Yes	1973	858
Test 4	$0.5 \cdot 10^{-1}$	1	$1.0 \cdot 10^{-6}$	Yes	198	220
Test 5	0.5	1	$1.0 \cdot 10^{-6}$	Yes	19	31
Test 6	$0.5 \cdot 10^{-4}$	0	$1.0 \cdot 10^{-6}$	Yes	193257	49
Test 7	$0.5 \cdot 10^{-3}$	0	$1.0 \cdot 10^{-6}$	Yes	19326	459
Test 8	$0.5 \cdot 10^{-2}$	0	$1.0 \cdot 10^{-6}$	Yes	1933	851
Test 9	$0.5 \cdot 10^{-1}$	0	$1.0 \cdot 10^{-6}$	Yes	194	215
Test 10	0.5	0	$1.0 \cdot 10^{-6}$	Yes	19	31
Test 11	$0.5 \cdot 10^{-2}$	1	$1.0 \cdot 10^{-10}$	Yes	1973	2181
Test 12	$0.5 \cdot 10^{-2}$	0	$1.0 \cdot 10^{-10}$	Yes	1933	2143

Table C.2. *Robustness check of the arc-length solver for 2DOF system in Figure C.2. All the test are done with $w = 0.25$.*

Benchmark Two

This benchmark example is based on an example in (Leon et al., 2011). In this article, several different nonlinear solution schemes are used to trace a 2 DOF non-linear function. The function exhibits both snap-back and snap-through behaviour for both degrees of freedom. In the problem considered, the external and internal forces are defined by Equation C.5.

$$\boldsymbol{f}_{ext} = \begin{bmatrix} 40 \\ 15 \end{bmatrix} \qquad \boldsymbol{f}_{int}(\boldsymbol{u}) = \begin{bmatrix} 10u_1 + 0.4u_2^3 - 5u_2^2 \\ 0.4u_1^3 - 3u_1^2 + 10u_2 \end{bmatrix} \tag{C.5}$$

The tangent stiffness is determined by differentiating the internal forces with respect to the degrees of freedom u_1 and u_2 as done in Equation C.6.

$$\boldsymbol{K}(\boldsymbol{u}) = \begin{bmatrix} 10 & 1.2u_2^2 - 10u_2 \\ 1.2u_1^2 - 6u_1 & 10 \end{bmatrix} \tag{C.6}$$

The programmed arc-length solver is again visually compared to the benchmark example. This is done in Figure C.5 and Figure C.6.

Again the similarity between the programmed arc-length solution in Figure C.5 and the solution from the benchmark example in Figure C.6 is satisfactory.

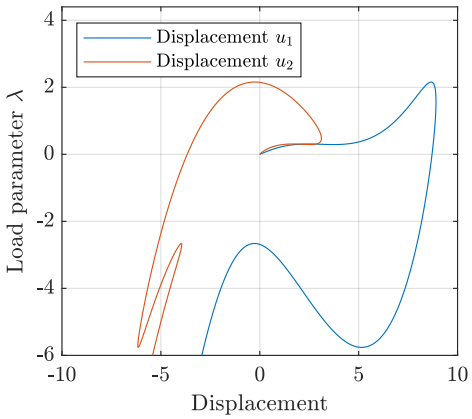


Figure C.5. *Arc-length solver solution for 2 DOF system. Solver setting are* $\Delta l = 0.011$, $\beta = 1$, *Residual tolerance = 0.001.*

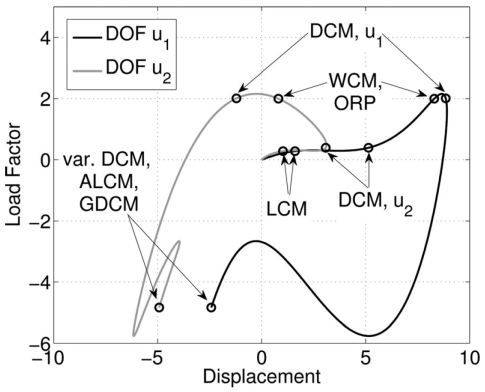


Figure C.6. *Benchmark example with load parameter* λ *plotted as a function of the displacement* u_1 *and* u_2. *(Leon et al., 2011)*

Algorithm for the Crisfield arc-length solver implemented in the MATLAB Framework

D

Algorithm 3 Crisfield arc-length solver

Save analysis data from last converged substep into new variables
Initialize vectors for $\{\delta \boldsymbol{D}^I\}$, $\{\delta \boldsymbol{D}^{II}\}$, $\{\delta \boldsymbol{D}\}$, $\{\Delta \boldsymbol{D}\}$ and scalars for $\delta\lambda$ and $\Delta\lambda$
Initialize iteration counter
if Substep = 1 **then**
    Update refinement of A-FNM elements
    Calculate and assemble the stiffness matrix and internal force vector
end if
Calculate displacement increment $\{\delta \boldsymbol{D}^I\}$ according to Equation 4.13
Calculate predictor value for load-level parameter $\Delta\lambda_1^n$ according to Equation 4.26
Calculate predictor displacement increment $\{\Delta \boldsymbol{D}\}_1$ according to Equation 4.22
Update displacement $\{\boldsymbol{D}\}_1^n = \{\boldsymbol{D}\}^{n-1} + \{\Delta \boldsymbol{D}\}_1^n$
Update load-level parameter $\lambda_1^n = \lambda^{n-1} + \Delta\lambda_1^n$
Update refinement of A-FNM elements
Calculate and assemble the stiffness matrix and internal force vector
Evaluate the residual force vector according to Equation 4.2
Update MPCs in the A-FNM elements
while The norm of the residual force is larger than the residual tolerance **do**
    Update iteration counter
    Calculate displacement correction $\{\delta \boldsymbol{D}^I\}$ and $\{\delta \boldsymbol{D}^{II}\}$ according to Equation 4.13
    Calculate constant, a_1, a_2 and a_3 according to Equation 4.20
    Solve Equation 4.19 and determine the correct load level parameter correction, $\delta\lambda$
    if No real roots are found in Equation 4.19 **then**
        Reset all analysis data to last converged substep,
        Bisect the step size, Δl,
        Exit solver
    end if
    Calculate displacement correction $\{\delta \boldsymbol{D}\}$ according to Equation 4.12.
    Update the incremental load-level parameter $\Delta\lambda$ according to Equation 4.15
    Update the incremental displacement vector $\{\Delta \boldsymbol{D}\}$ according to Equation 4.14
    Update displacement $\{\boldsymbol{D}\}_{i+1}^n = \{\boldsymbol{D}\}^{n-1} + \{\Delta \boldsymbol{D}\}_{i+1}^n$
    Update load-level parameter $\lambda_{i+1}^n = \lambda^{n-1} + \Delta\lambda_{i+1}^n$
    Update refinement of A-FNM elements
    Calculate and assemble the stiffness matrix and internal force vector
    Evaluate the residual force vector according to Equation 4.2
    if Residual error or number or iterations exceeds a predefined value **then**
        Reset all analysis data to last converged substep,
        Bisect the step size, Δl,
        Exit solver
    end if
    Update MPCs in the A-FNM elements
end while
Save all analysis data
Exist Crisfields arc-length solver

Derivative of Displacement Jump Vector

This appendix determines the full expression for each of the terms in, $\left\{\frac{\partial\{\boldsymbol{s}_L\}}{\partial d_j}\right\}$, that is needed in order to define $\boldsymbol{h}$, in Equation 5.7. This is required to implement the local energy arc-length solver in the MATLAB Framework.

An expression for the displacement jump vector in local coordinates is defined in Equation 5.32 and repeated here to ease the reading.

$$\begin{aligned}\{\boldsymbol{s}_L\} &= \begin{bmatrix}\nu_{1,\text{norm}} & -\nu_{2,\text{norm}} \\ \nu_{2,\text{norm}} & \nu_{1,\text{norm}}\end{bmatrix}\begin{bmatrix}N_1u_1^+ - N_1u_1^- + N_2u_2^+ - N_2u_2^- \\ N_1v_1^+ - N_1v_1^- + N_2v_2^+ - N_2v_2^-\end{bmatrix} \\ &= \begin{bmatrix}\nu_{1,\text{norm}}[N_1u_1^+ - N_1u_1^- + N_2u_2^+ - N_2u_2^-] - \nu_{2,\text{norm}}[N_1v_1^+ - N_1v_1^- + N_2v_2^+ - N_2v_2^-] \\ \nu_{2,\text{norm}}[N_1u_1^+ - N_1u_1^- + N_2u_2^+ - N_2u_2^-] + \nu_{1,\text{norm}}[N_1u_1^+ - N_1u_1^- + N_2u_2^+ - N_2u_2^-]\end{bmatrix}\end{aligned} \tag{E.1}$$

The final expression for the derivatives of the displacement jump vector are lengthy and in order to keep the overview, two new functions are defined. These are listed below in Equation E.2.

$$\begin{aligned}f_u(\boldsymbol{d}) &= [N_1u_1^+ - N_1u_1^- + N_2u_2^+ - N_2u_2^-] \\ f_v(\boldsymbol{d}) &= [N_1v_1^+ - N_1v_1^- + N_2v_2^+ - N_2v_2^-]\end{aligned} \tag{E.2}$$

By introducing these function, the displacement jump vector can be expressed by Equation E.3, where both $\nu_{1,\text{norm}}(\boldsymbol{d})$, $\nu_{2,\text{norm}}(\boldsymbol{d})$, $f_u(\boldsymbol{d})$ and $f_v(\boldsymbol{d})$ are functions of the degrees of freedom in the model.

$$\{\boldsymbol{s}_L\} = \begin{bmatrix}\nu_{1,\text{norm}}(\boldsymbol{d})f_u(\boldsymbol{d}) - \nu_{2,\text{norm}}(\boldsymbol{d})f_v(\boldsymbol{d}) \\ \nu_{2,\text{norm}}(\boldsymbol{d})f_u(\boldsymbol{d}) + \nu_{1,\text{norm}}(\boldsymbol{d})f_v(\boldsymbol{d})\end{bmatrix} \tag{E.3}$$

Differentiation of the displacement jump $\{\boldsymbol{s}_L\}$ vector is done by use of the product rule, as defined in Equation E.4.

$$\left\{\frac{\partial\{\boldsymbol{s}_L\}}{\partial d_j}\right\} = \begin{bmatrix}\frac{\partial\nu_{1,\text{norm}}(\boldsymbol{d})}{\partial d_j}f_u(\boldsymbol{d}) + \nu_{1,\text{norm}}(\boldsymbol{d})\frac{\partial f_u(\boldsymbol{d})}{d_j} - \frac{\partial\nu_{2,\text{norm}}(\boldsymbol{d})}{\partial d_j}f_v(\boldsymbol{d}) + \nu_{2,\text{norm}}(\boldsymbol{d})\frac{\partial f_v(\boldsymbol{d})}{d_j} \\ \frac{\partial\nu_{2,\text{norm}}(\boldsymbol{d})}{\partial d_j}f_u(\boldsymbol{d}) + \nu_{2,\text{norm}}(\boldsymbol{d})\frac{\partial f_u(\boldsymbol{d})}{d_j} + \frac{\partial\nu_{1,\text{norm}}(\boldsymbol{d})}{\partial d_j}f_v(\boldsymbol{d}) + \nu_{1,\text{norm}}(\boldsymbol{d})\frac{\partial f_v(\boldsymbol{u})}{d_j}\end{bmatrix} \tag{E.4}$$

Each of the terms in the above equation are now defined. First the derivatives of $f_u(\boldsymbol{d})$ and $f_v(\boldsymbol{d})$ are determined in Equation E.5:

$$\begin{aligned}&\frac{\partial f_u}{\partial u_1^+} = N_1 \quad \frac{\partial f_u}{\partial u_1^-} = -N_1 \quad \frac{\partial f_u}{\partial u_2^+} = N_2 \quad \frac{\partial f_u}{\partial u_2^-} = -N_2 \\ &\frac{\partial f_v}{\partial v_1^+} = N_1 \quad \frac{\partial f_v}{\partial v_1^-} = -N_1 \quad \frac{\partial f_v}{\partial v_2^+} = N_2 \quad \frac{\partial f_v}{\partial v_2^-} = -N_2 \\ &\frac{\partial f_u}{\partial v_1^+} = \frac{\partial f_u}{\partial v_1^-} = \frac{\partial f_u}{\partial u_2^+} + \frac{\partial f_u}{\partial v_2^-} = 0 \quad \frac{\partial f_v}{\partial u_1^+} = \frac{\partial f_v}{\partial u_1^-} = \frac{\partial f_v}{\partial u_2^+} + \frac{\partial f_v}{\partial u_2^-} = 0\end{aligned} \tag{E.5}$$

And the terms in $\frac{\partial \nu_{1,\mathrm{norm}}(\boldsymbol{d})}{\partial d_j}$ and $\frac{\partial \nu_{2,\mathrm{norm}}(\boldsymbol{d})}{\partial d_j}$ are defined in Equation E.6 through Equation E.13.

$$\frac{\partial \nu_{1,norm}}{\partial u_1^+} = \frac{\partial \nu_{1,norm}}{\partial u_1^-} = \frac{T_{\nu_1,u}}{N} \tag{E.6}$$

$$\frac{\partial \nu_{1,norm}}{\partial u_2^+} = \frac{\partial \nu_{1,norm}}{\partial u_2^-} = -\frac{T_{\nu_1,u}}{N} \tag{E.7}$$

$$\frac{\partial \nu_{1,norm}}{\partial v_1^+} = \frac{\partial \nu_{1,norm}}{\partial v_1^-} = \frac{T_{\nu_1,v}}{N} \tag{E.8}$$

$$\frac{\partial \nu_{1,norm}}{\partial v_2^+} = \frac{\partial \nu_{1,norm}}{\partial v_2^-} = -\frac{T_{\nu_1,v}}{N} \tag{E.9}$$

$$\frac{\partial \nu_{2,norm}}{\partial u_1^+} = \frac{\partial \nu_{2,norm}}{\partial u_1^-} = \frac{T_{\nu_2,u}}{N} \tag{E.10}$$

$$\frac{\partial \nu_{2,norm}}{\partial u_2^+} = \frac{\partial \nu_{2,norm}}{\partial u_2^-} = -\frac{T_{\nu_2,u}}{N_{\nu_2,u}} \tag{E.11}$$

$$\frac{\partial \nu_{2,norm}}{\partial v_1^+} = \frac{\partial \nu_{2,norm}}{\partial v_1^-} = \frac{T_{\nu_2,v}}{N_{\nu_2,v}} \tag{E.12}$$

$$\frac{\partial \nu_{2,norm}}{\partial v_2^+} = \frac{\partial \nu_{2,norm}}{\partial v_2^-} = -\frac{T_{\nu_2,v}}{N_{\nu_2,v}} \tag{E.13}$$

where:

$$T_{\nu_1,u} = -(y_1^- + y_1^+ - y_2^- - y_2^+ + v_1^- + v_1^+ - v_2^- - v_2^+)^2 \tag{E.14}$$

$$\begin{aligned} T_{\nu_1,v} = (&u_1^- + u_1^+ - u_2^- - u_2^+ + x_1^- + x_1^+ - x_2^- - x_2^+) \\ &(v_1^- + v_1^+ - v_2^- - v_2^+ + y_1^- + y_1^+ - y_2^- - y_2^+) \end{aligned} \tag{E.15}$$

$$\begin{aligned} T_{\nu_2,u} = (&y_1^+ + y_1^- + v_1^+ + v_1^- - y_2^+ - y_2^- - v_2^+ - v_2^-) \\ &(u_1^- + u_1^+ - u_2^- - u_2^+ + x_1^- + x_1^+ - x_2^- - x_2^+) \end{aligned} \tag{E.16}$$

$$T_{\nu_2,v} = -(u_1^- + u_1^+ - u_2^- - u_2^+ + x_1^- + x_1^+ - x_2^- - x_2^+)^2 \tag{E.17}$$

and

$$\begin{aligned} N = \Big(&(u_1^-)^2 + (2u_1^+ - 2u_2^- - 2u_2^+ + 2x_1^- + 2x_1^+ - 2x_2^- \\ &-2x_2^+)u_1^- + (u_1^+)^2 + (2x_1^+ - 2x_2^- - 2x_2^+ - 2u_2^- - 2u_2^+ \\ &+2x_1^-)u_1^+ + (u_2^-)^2 + (-2x_1^+ + 2x_2^- + 2x_2^+ + 2u_2^+ - 2x_1^-)u_2^- \\ &+(u_2^+)^2 + (-2x_1^+ + 2x_2^- + 2x_2^+ - 2x_1^-)u_2^+ + (v_1^-)^2 + (2v_1^+ - 2v_2^- \\ &-2v_2^+ + 2y_1^- + 2y_1^+ - 2y_2^- - 2y_2^+)v_1^- + (v_1^+)^2 + (2y_1^- + 2y_1^+ \\ &-2y_2^- - 2y_2^+ - 2v_2^- - 2v_2^+)v_1^+ + (v_2^-)^2 + (-2y_1^- - 2y_1^+ + 2y_2^- \\ &+2y_2^+ + 2v_2^+)v_2^- + (v_2^+)^2 + (-2y_1^- - 2y_1^+ + 2y_2^- + 2y_2^+)v_2^+ \\ &+(x_1^-)^2 + (2x_1^+ - 2x_2^- - 2x_2^+)x_1^- + (x_1^+)^2 + (-2x_2^- - 2x_2^+)x_1^+ \\ &+(x_2^-)^2 + 2x_2^+x_2^- + (x_2^+)^2 + (y_1^- + y_1^+ - y_2^- - y_2^+)^2 \Big)^{3/2} \end{aligned} \tag{E.18}$$

Algorithm for the local energy arc-length solver implemented in the MATLAB Framework

F

Algorithm 4 Local energy arc-length solver

Save analysis data from last converged substep into new variables
Initialize vectors for $\{\delta \boldsymbol{D}\}$, $\{\Delta \boldsymbol{D}\}$ and scalars for $\delta\lambda$ and $\Delta\lambda$
Initialize the amount of dissipated energy, ΔQ, to zero
Initialize iteration counter
Calculate and assemble the stiffness matrix, internal force vector and constraint derivatives, $\{\boldsymbol{h}\}_i^n$
Calculate the path-following constraint, g_i^n according to Equation 5.5
Assemble the system of equations defined in Equation 4.8 and solve for the load-level parameter correction $\delta\lambda_i^n$ and the displacement correction $\{\delta \boldsymbol{D}\}_i^n$
Update displacement $\{\boldsymbol{D}\}_1^n = \{\boldsymbol{D}\}^{n-1} + \{\delta \boldsymbol{D}\}_1^n$
Update load-level parameter $\lambda_1^n = \lambda^{n-1} + \delta\lambda_1^n$
Update refinement of A-FNM elements
Calculate and assemble the stiffness matrix, internal force vector and constraint derivatives, $\{\boldsymbol{h}\}_i^n$
Evaluate the residual force vector according to Equation 4.2
Update MPCs in the A-FNM elements
while The norm of the residual force is larger than the residual tolerance **do**
    Update iteration counter
    Calculate the path-following constraint, g_i^n according to Equation 5.5
    Assemble the system of equations defined in Equation 4.8 and solve for the load-level parameter correction $\delta\lambda_i^n$ and the displacement correction $\{\delta \boldsymbol{D}\}_i^n$
    Update the incremental load-level parameter $\Delta\lambda$ according to Equation 4.15
    Update the incremental displacement vector $\{\Delta \boldsymbol{D}\}$ according to Equation 4.14
    Update displacement $\{\boldsymbol{D}\}_{i+1}^n = \{\boldsymbol{D}\}^{n-1} + \{\Delta \boldsymbol{D}\}_{i+1}^n$
    Update load-level parameter $\lambda_{i+1}^n = \lambda^{n-1} + \Delta\lambda_{i+1}^n$
    Update refinement of A-FNM elements
    Calculate and assemble the stiffness matrix, internal force vector and constraint derivatives, $\{\boldsymbol{h}\}_i^n$
    Evaluate the residual force vector according to Equation 4.2
    if Residual error or number or iterations exceeds a predefined value **then**
        Reset all analysis data to last converged substep,
        Bisect the step size, Δl,
        Exit solver
    end if
    Update MPCs in the A-FNM elements
end while
Save all analysis data
Exist local energy arc-length solver

Plot of Local Energy Arc-Length Solver Solution

G

A plot of the solution obtained with the local energy arc-length solver explained in chapter 5, is illustrated in this appendix. The solution is done for the same problem as is used in section 4.2. The arc-length solver is run with Crisfield arc-length solver for the first 19 substep to initiate the solver, and then it changes to the local energy arc-length solver. The arc-length solver is run with a step-size of $\Delta\tau = 0.0001$ and a residual of 1×10^{-4} N. The solution is compared with a displacement controlled solution. Further, the analysis is stopped before the full solution is obtained due to large amount of time needed to finish the analysis. However, as illustrated in Figure G.1, the arc-length method traces the displacement controlled solution satisfactory and do also capture the snap-through and snap-back behaviour.

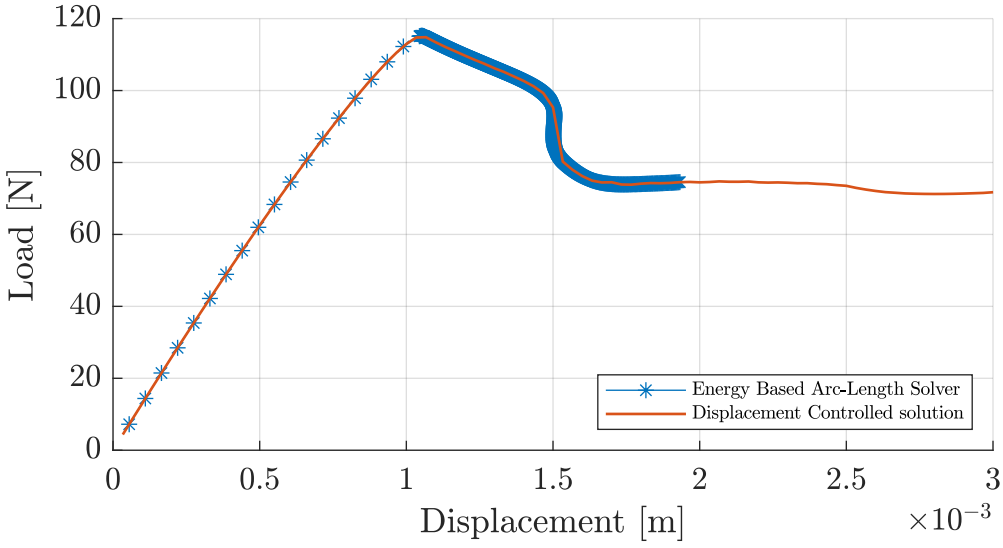


Figure G.1. *Validation of local energy arc-length solver. Solver setting are* $\Delta\tau = 0.0001$ *and a residual error of* 1×10^{-4} N.

Algorithm for the global energy arc-length solver implemented in the MATLAB Framework

H

Algorithm 5 Global energy arc-length solver

Save analysis data from last converged substep into new variables
Initialize vectors for $\{\delta \boldsymbol{D}\}$, $\{\Delta \boldsymbol{D}\}$ and scalars for $\delta\lambda$ and $\Delta\lambda$
Initialize iteration counter
Calculate and assemble the stiffness matrix and internal force vector
Calculate the constraint derivatives, $\{\boldsymbol{h}\}_i^n$ according to Equation 5.46
Calculate the constraint derivatives, w_i^n according to Equation 5.47
Calculate the path-following constraint, g_i^n according to Equation 5.45
Assemble the system of equations defined in Equation 4.8 and solve for the load-level parameter correction $\delta\lambda_i^n$ and the displacement correction $\{\delta \boldsymbol{D}\}_i^n$
Update displacement $\{\boldsymbol{D}\}_1^n = \{\boldsymbol{D}\}^{n-1} + \{\delta \boldsymbol{D}\}_1^n$
Update load-level parameter $\lambda_1^n = \lambda^{n-1} + \delta\lambda_1^n$
Update refinement of A-FNM elements
Calculate and assemble the stiffness matrix and internal force vector
Evaluate the residual force vector according to Equation 4.2
Update MPCs in the A-FNM elements
while The norm of the residual force is larger than the residual tolerance **do**
    Update iteration counter
    Calculate the constraint derivatives, $\{\boldsymbol{h}\}_i^n$ according to Equation 5.46
    Calculate the constraint derivatives, w_i^n according to Equation 5.47
    Calculate the path-following constraint, g_i^n according to Equation 5.45
    Assemble the system of equations defined in Equation 4.8 and solve for the load-level parameter correction $\delta\lambda_i^n$ and the displacement correction $\{\delta \boldsymbol{D}\}_i^n$
    Update the incremental load-level parameter $\Delta\lambda$ according to Equation 4.15
    Update the incremental displacement vector $\{\Delta \boldsymbol{D}\}$ according to Equation 4.14
    Update displacement $\{\boldsymbol{D}\}_{i+1}^n = \{\boldsymbol{D}\}^{n-1} + \{\Delta \boldsymbol{D}\}_{i+1}^n$
    Update load-level parameter $\lambda_{i+1}^n = \lambda^{n-1} + \Delta\lambda_{i+1}^n$
    Update refinement of A-FNM elements
    Calculate and assemble the stiffness matrix and internal force vector
    Evaluate the residual force vector according to Equation 4.2
    if Residual error or number or iterations exceeds a predefined value **then**
        Reset all analysis data to last converged substep,
        Bisect the step size, Δl,
        Exit solver
    end if
    Update MPCs in the A-FNM elements
end while
Save all analysis data
Exist global energy arc-length solver

Virtual Clip Gauge Scripts

I

In the following the general layout of the *ImageRegistation* and *Virtual Clip Gauge* is shown in Table I.1 and I.2 respectively.

1. Initialize
Start with the top detail
3. Check if region of interest for the top detail is within the image
If Yes: Go to step 4, else go to step 16
4. Use the SURF algorithm with a region of interest to the following:
5. Detect local feature in both the detail and the image
6. Extract local feature descriptors
7. Find candidate matches in detail and image
8. Retrieve locations in detail and image for the candidate matches
9. Check if enough candidate matches are found. If yes continue else go to step 12
10. Remove outliers and determine the geometric transformation
11. Check if enough matching points have been found in step 10. If yes go to step 24 else continue
12. Use BRISK algorithm with a region of interest and the procedure in step 5-10 combined with the candidate matches with those from step 4
13. Check if enough matching points have been found. If yes go to step 24 else continue
14. Use KAZE algorithm with a region of interest and the procedure in step 5-10 combined with the candidate matches with those from step 4 and 12
15. Check if enough matching points have been found. If yes go to step 24 else go to step 20
16. Use SURF algorithm without a region of interest and the procedure in step 5-10
17. Check if enough matching points have been found. If yes go to step 24 else continue
18. Use BRISK algorithm without a region of interest and the procedure in step 5-10 combined with the candidate matches with those from step 16
19. Check if enough matching points have been found. If yes go to step 24 else continue
20. Use SIFT algorithm without a region of interest and the procedure in step 5-10 combined with the candidate matches with those from step 16 and 18
21. Check if enough matching points have been found. If yes go to step 24 else continue
22. Use KAZE algorithm without a region of interest and the procedure in step 5-10 combined with the candidate matches with those from step 16, 18 and 20
23. Check if enough matching points have been found. If yes go to step 24 else continue
Repeat step 3-23 for the bottom detail. Reuse the local features found on the image from step 16, 18, 20 and 22 if possible
24. Return location, rotation and number of matching points found for the two details to the *Virtual Clip Gauge*

Table I.1. *ImageRegistration function.*

1. Start the Virtual Clip Gauge.
2. Select folder, containing data, camera parameters and images from the experiment, add paths for these.
3. Load camera parameters: used to undistort images when loading them.
4. Create results folder.
5. Read image names.
6. Select Search Area: Loads first and last image, then select the search area based on the crosses.
7. Calibration: calculate the distance per pixel using a calibration function made by the authors.
8. Load first image and select top and bottom detail.
9. Save position and size of the details.
10. Check the selected details.
11. Load first image.
12. Run *ImageRegistration* with the two details on the first image.
13. Calculate the center position of the found details.
14. Set region of interest based on the center position found in step 13.
15. Initiate counter $count = 0$
16. **for loop** - loop over the remaining number of images.
17. Load image.
18. Run ImageRegistration with the two details.
19. Check if the details are found. If Yes: got to step 20 else report error and go to step 24.
20. Check if the angles calculated in step 18 have increased by five degrees.
If Yes: Update both details else go to step 25
21. Update detail positions.
22. Calculate Center positions.
23. Set region of interest based on the center positions.
24. Update counter $count = 0$
25. **for loop** - loop over the number of images.
26. Calculate distance in pixels between the center positions.
27. Calculate distance in mm using the distance in pixel and the calibration from step 7.
28. Manually select a cutoff point and remove outliers using the Matlab function *filloutliers*.
29. Calculate the frame rate of the camera based on the timestamp on the first and last image.
30. Calculate the CMOD by subtracting the first distance found in step 27 from the subsequent distances.
31. Calculate a time for each CMOD measurement by $\frac{1}{\text{Frame rate}} \cdot$ index in CMOD
33. Load data set from the experiment and synchronize the time with the clip gauge.
34. Save the results from the Virtual Clip Gauge.

Table I.2. *Virtual Clip Gauge Script.*